VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

VYUŽITÍ STROJOVÉHO UČENÍ VE FIREMNÍM PROSTŘEDÍ

THE USE OF MACHINE LEARNING IN BUSINESS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Helena Balarinová

AUTHOR

VEDOUCÍ PRÁCE — Ing. Jiří Kříž, Ph.D.

SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav informatiky
Studentka:	**Bc. Helena Balarinová**
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	**Ing. Jiří Kříž, Ph.D.**
Akademický rok:	2019/20

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Využití strojového učení ve firemním prostředí

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem je analýza aktuálního stav organizace, jejich produktů a využití dat. Hlavní cíl práce je na základě dat pomocí strojového učení získat cenné informace pro budoucí vývoj firmy.

Základní literární prameny:

BEN-GAN, Itzik, Dejan SARKA and Ron TALMAGE. Querying Microsoft SQL Server 2012: Exam 70-461 Training Kit. 1st Edition. California: Microsoft, 2012. ISBN 978-0-7356-6605-4.

LABERGE, Robert. Datové sklady: agilní metody a business intelligence. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

LEONARD, Andy. SQL Server 2012 Integration Services Design Patterns. 1st Edition. California: Apress, 2012. ISBN 978-1430237716.

MULLER, Andreas C. a Sarah GUIDO. Introduction to machine learning with Python: a guide for data scientists. 1 st Edition. Sebastopol, CA: O'Reilly Media, 2016. ISBN 9781449369415.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně dne 29.2.2020

L. S.

doc. RNDr. Bedřich Půža, CSc. ředitel	doc. Ing. et Ing. Stanislav Škapa, Ph.D. děkan

Abstrakt

Diplomová práce je zaměřena na využití strojového učení ve firemním prostředí. Součástí práce je podrobný postup, jak jsem došla k cílovým predikcím, včetně očištění a práci s daty.

Abstract

The master‘s thesis is focused at the use of machine learning in business enviroment. In this thesis I describe detailed process of how I found the target prediction, this includes cleaning data and work with them.

Klíčová slova

strojové učení, Business Intelligence, datové sklady, Python

Key words

machine learning, Business Intelligence, data warehouse, Python

Bibliografická citace

BALARINOVÁ, Helena. *Využití strojového učení ve firemním prostředí* [online]. Brno, 2020 [cit. 2020-05-31]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/127654. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jiří Kříž.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil/a autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 31. května 2020

_____

podpis studenta

Poděkování

Mé poděkování patří především panu Ing. Jiřímu Křížovi, Ph.D. za jeho odborné rady a vedení diplomové práce. Rovněž bych chtěla poděkovat firmě Solitea Česká republika a.s., že mi umožnila využít firemní data pro napsání diplomové práce.

OBSAH

ÚVOD

V dnešní době roste čím dál více trend umět dobře využívat firemní data. Ve firmách se data z databází využívají pro různé analýzy, vizualizaci dat, díky kterým je můžeme lépe pochopit, ale všechny tyto věci se začínají zdát poněkud nedostačující. Firmy chtějí více než jen čísla svých aktuálních prodejů a líbivé grafy s nárůstem zákazníků. Mají zájem poznat data, která uchovávají, ještě blíže a díky nim získat konkurenční převahu na trhu. Chtějí znát budoucnost. Chtějí vědět, jak se jejich zákazníci mohou v budoucnu chovat. Chtějí být o krok napřed před ostatními. Na řadu přichází pojmy jako je umělá inteligence, strojové učení nebo hluboké učení. Zastavím se u pojmu strojového učení, které bude provázet celou mou práci. Algoritmy strojového učení se dokážou nad firemními daty učit a z nich vyvozovat další závěry. Není to pouhý pohled na historická data a kopírování křivky, ale jde o nalezení spojitosti mezi daty, díky kterým poznáme, které aspekty nás v budoucnu mohou ovlivnit a které ne.

V mé diplomové práci budu aplikovat strojové učení ve společnosti Solitea Česká republika a.s. Zaměřím se na jeden z jejich produktů a budu předpovídat jeho prodeje anebo také zjišťovat, za jak dlouho může zákazník přejít z nižšího tarifu produktu na vyšší. Základem je dobře znát svá data a umět odhadnout, které aspekty mohou tyto skutečnosti ovlivňovat a které nikoliv.

Začátek mé práce budu věnovat teoretickým východiskám, která se budou hodit pro lepší pochopení následujících kapitol. Následující část se bude věnovat analýze současného stavu ve firmě, kde provedu různé analýzy, které mi pomohou blíže poznat firmu a vyvodit z nich slabé a silné stránky. Okrajově se zaměřím také na samotný produkt, kterému se budu hlavně věnovat.

Hlavní část práce je samotný vlastní návrh řešení, kde budu řešit dva modely strojového učení. Celý návrh vlastního řešení pojmu jako projekt, takže se v této části objeví také časová analýza nebo analýza rizik.

Na konci mé práce samozřejmě nebude chybět ekonomické zhodnocení a zlepšení, které má řešení přinesou.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Cílem této diplomové práce je začít využívat strojové učení ve firemním prostředí a díky vytvořeným modelům získat přesnější predikce do budoucna. Firma disponuje velkým množstvím dat, takže byla jen otázka času, kdy projeví zájem o jejich lepší využití než jen v reportovacích nástrojích.

Před vytvářením návrhů jsem se seznámila s daty a produktem iDoklad. Zkoumala jsem, jaké funkcionality jednotlivé tarify nabízí a které jsou nejvíce využívané. Dále jsem zjišťovala, kde a jak je v databázi uvedeno, který tarif aktuálně zákazník využívá a jaké byly jeho předcházející. Z toho vychází i to, jestli je ten zákazník nový anebo stávající. Dále jsem se věnovala celkové analýze firmy. Ta spočívala v tom, že jsem provedla analýzu vnitřního (analýza 7S) i vnějšího prostředí (SLEPT, Porter). Na základě těchto analýz jsem pak mohla definovat silné a slabé stránky firmy.

Díky poznatkům, které jsem načerpala, jsem mohla začít přemýšlet, které informace z dat by pro firmu mohly být zajímavé. Dospěla jsem k názoru, že by bylo dobré vědět, kdy zákazník bude mít zájem přejít na vyšší tarif. Spojitosti jsem hledala mezi funkcionalitami, které zákazník nejvíce využíval. Model se učil, které funkcionality mají pro konkrétního zákazníka největší význam a ovlivní ho, aby si předplatil vyšší tarif. Druhý prediktivní model, který si myslím, že by mohl mít ve firmě využití je běžná predikce budoucích prodejů. Tyto informace jsou zajímavé pro každého manažera, a jelikož některé tarify je možné předplatit si až na rok dopředu, víme, kdo naši službu bude nadále využívat a predikování je jednodušší.

Software, který budu využívat při praktické části je primárně Microsoft SQL Management Studio, díky kterému se připojím k našemu firemnímu datovému skladu, ze kterého budu čerpat data. Pro další práci s daty využiji prostředí PyCharm, ve kterém budu používat programovací jazyk Python a platformu Anaconda pro datovou analýzu. Jeden z mých prediktivních modelů budu vytvářet v Microsoft Azure Machine Learning Studio, což je intuitivní nástroj pro vytváření modelů strojového učení. V neposlední řadě znázorním výsledky svých predikcí v reportovacím nástroji Microsoft Power BI.

1 TEORETICKÁ VÝCHODISKA

1.1 Databáze

Databáze je soubor strukturovaných dat. Můžeme si to představit na příkladu z reálného světa. **Entita** (tabulka) je prvek z reálného světa (např. člověk, město) a je popsán nějakými vlastnostmi, kterým se říká **atribut** (sloupec) (u člověka např. jméno, příjmení, věk). Více entit má mezi sebou určitý vztah, to se pojmenovává jako vazba mezi entitami. Vazba může být 1:1, kdy každý člověk má jedno pohlaví. Další typ je vazba 1:N, kdy jeden člověk může mít více kreditních karet, ale naopak kreditní karta může být vlastněna pouze jedním člověkem. Poslední typ se nazývá M:N, kdy jeden student může mít zapsáno více předmětů a předmět může být také zapsán více studenty. V dnešní době se setkáváme s **relačními databázemi**. Relace je tabulka, která se skládá ze sloupců a řádků. Prvek relace je jeden konkrétní řádek v tabulce. V databázové tabulce si musíme zvolit jeden **primární klíč**, který nabývá v celé tabulce jedinečné hodnoty (1).

Pro práci v databázi se používá dotazovací jazyk **SQL** – Structured Query Language. Jazyk využívá nástroje pro tvorbu databází, tabulek a na práci s manipulaci s daty (1).

T-SQL je hlavní jazyk, který se používá k manipulaci dat v Microsoft SQL Serveru. T-SQL je nadstavba klasického dotazovacího jazyka SQL, který je standardem schválený organizací ISO. (2, str. 3)

1.2 Datové sklady

Informační systémy, které se nachází ve většině organizacích zpracovávají transakce a ukládají se tedy do transakčních systémů (**OLTP** – Online Transaction Processing). Využívají se především pro vkládání, mazání dat. Tento systém ukládání dat funguje dobře k tomu, k čemu byly implementovány, např. evidence zákazníků, vystavování objednávek a další. Také z těchto systémů vzniká velké množství dat, které mohou nést cenné informace. Pokud bychom ale chtěli analyzovat data z těchto systémů bylo by to značně problematické. Z toho důvodu se využívají datové sklady (3).

Transakční databáze

- nedostatečná historie dat (3),
- není možné zpracovávat data z jiných aplikací (3),
- zátěž na provozní systém (3).

Datový sklad

- integrace dat z více aplikací (3),
- datový sklad je fyzicky i logicky oddělen od provozních systémů (3),
- obsahuje historická data (3),
- data se periodicky načítají z provozních systémů (3).

1.2.1 ETL proces

Data z provozních systémů se načítají do datového skladu v čase, kdy nejsou provozní systémy moc zatíženy. Při samotném plnění se používají tři kroky (3):

E (Extract) – Extrakce vstupních dat – výběr dat pomocí různých metod (3).

T (Transformation) – Transformace vstupních dat – čištění dat, validace dat, integrace a časové označení dat (3).

L (Load) – Načtení dat do datového skladu (3).

Na začátku každého projektu datového skladu se provádí počáteční načítání dat. Po počátečním načtení dat přichází na řadu inkrementální načítání. Rozdíl je v tom, že data v datovém skladu už jsou a každé nové načítání potřebujeme rozlišit, zda je řádek nový anebo se od předchozího importu změnil. Pokud se něco změnilo, spustí se proces, který určí dopad změny. (4, str. 247)

1.2.2 Tabulky faktů a dimenzí

V datovém skladu je možné se setkat s dvěma druhy tabulek – fakta a dimenze. Toto členění je z toho důvodu, že data se musí strukturovat do schémat kvůli přehlednosti a lepší orientaci na uživatele (5).

Tabulky faktů

Tabulka faktů obsahuje z celého datového skladu nejvíce záznamů. Uvnitř se nachází měřitelná data, nad kterými je možné používat agregované funkce (5).

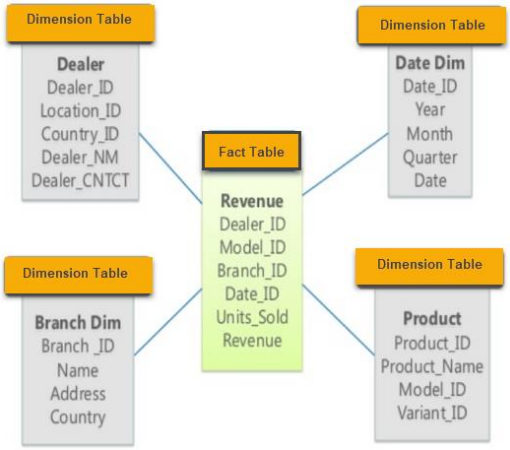
Tabulka dimenzí

Oproti tabulce faktů neuchovávají tabulky dimenzí žádná měřitelná data. Slouží k tomu, aby poskytly tabulce faktů potřebné informace. Při připojení časové tabulky dimenzí získáme časové informace a zjistíme k jakému časovému bodu jsou data ve faktové tabulce uložená (5).

1.2.3 Schéma datového skladu

Hvězda

Ve středu hvězdicového schématu se nachází tabulka faktů, ve které je obsažen cizí klíč do tabulky dimenzí. Z tabulky faktů vychází tabulky dimenzí, které mají každá svůj vlastní primární klíč (6).

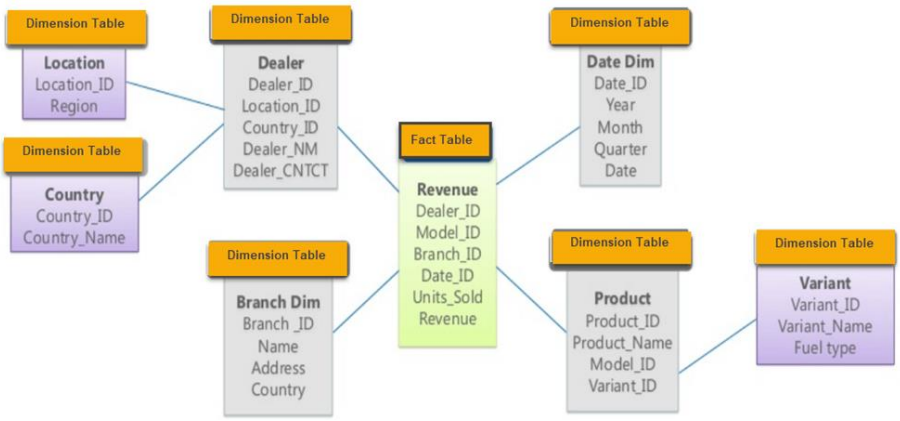


Obrázek 1: Schéma hvězda

(Zdroj: 6)

Sněhová vločka

Tabulka faktů se nachází stejně jako u hvězdicového schématu uprostřed a opět z ní vychází tabulky dimenzí, ty jsou dále normalizované do dalších tabulek. Toto uspořádání je vhodné pro rychlost čtení dat (6).

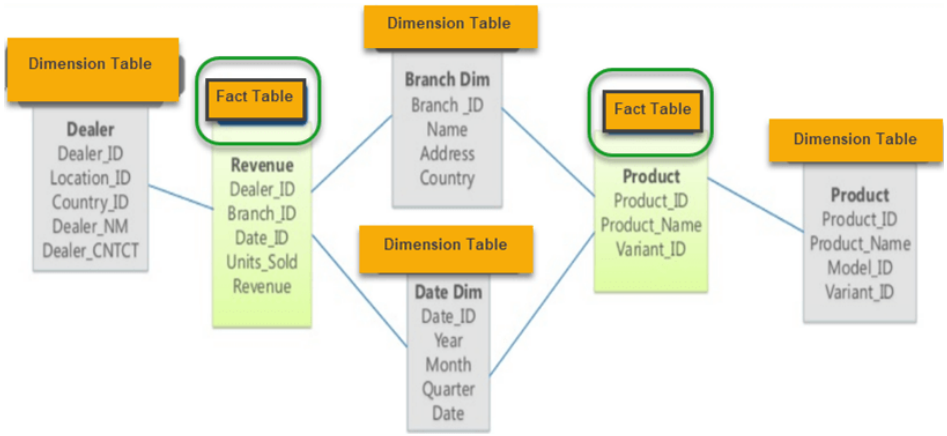


Obrázek 2: Schéma vločka

(Zdroj: 6)

Souhvězdí

Schéma souhvězdí obsahuje dvě tabulky faktů, které mají společné tabulky dimenzí mezi nimi (6).



Obrázek 3: Schéma souhvězdí

(Zdroj: 6)

1.3 SQL Server Integration Services

V SQL Server Integration Services je možné provádět ETL proces. Je možné vytvářet datové integrace pomocí SSIS balíčku (packages). Není nutné použít jeden balíček na všechny tři fáze ETL procesu, ale můžeme si vytvořit každý balíček zvlášť na jednotlivé fáze. Pro vytváření balíčku se používá software, který se podobá Visual Studiu – SSDT, SQL Server Data Tools (7).

1.4 Business Intelligence

Business Intelligence se využívá pro ucelenou a efektivní práci s daty. Slouží pro zpracování historických dat, predikcí dat anebo simulaci budoucího vývoje. „*Jejich hlavním cílem je poskytnout kvalitní data pro rychlejší a efektivnější rozhodování.“ (8)*

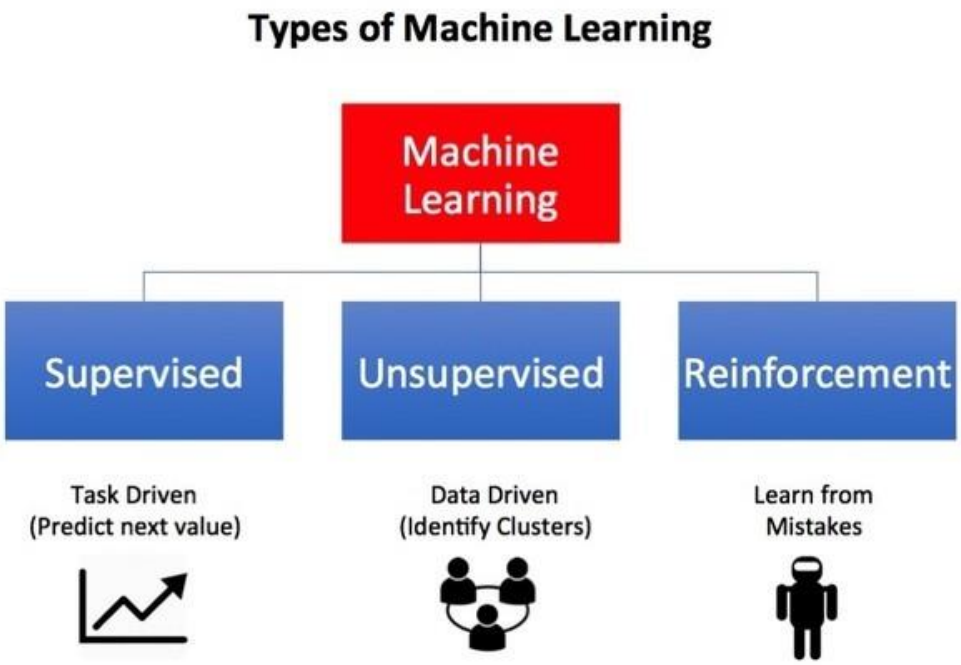
Systém datového skladu je uživatelem chápan podobně jako systém BI, ale rozdíl je v tom, že systém datového skladu se nezaměřuje na zájmy podnikového využití. Tím pádem se Business Intelligence stává nedílnou součástí systému datového skladu. Aby byl systém BI úspěšný musí vycházet z podnikových potřeb. Základním kamenem projektu jsou kvalitní data. Důležité je, zajistit úzkou spolupráci mezi IT oddělením, vývojáři a podnikovou jednotkou, tím může být manažer ve firmě, který využívá BI systém anebo zákazník. (4, str. 27)

1.5 Machine learning

Machine learning v překladu strojové učení se začalo objevovat v devadesátých letech a definoval ho Arthur Samuel. Hlavní záměr využití strojového učení vychází z toho, že v dnešním světě máme velké množství dat a je skoro nemožné porozumět všem. Více než 80 % dat je nestrukturovaných, například audio, video, fotografie, dokumenty, grafy a další. Najít souvislosti mezi těmito daty je nemožné pro lidský mozek. Množství dat je obrovské a procházení dat by zabralo nepřiměřené množství času, proto tady přichází na scénu právě strojové učení. Zvládne zpracovat velké množství dat za minimum času a také modely, které vytvoří mají vysokou přesnost. Díky tomuto může organizace dříve identifikovat příležitosti, které jsou ziskové, a naopak se vyhnout neznámým rizikům (9).

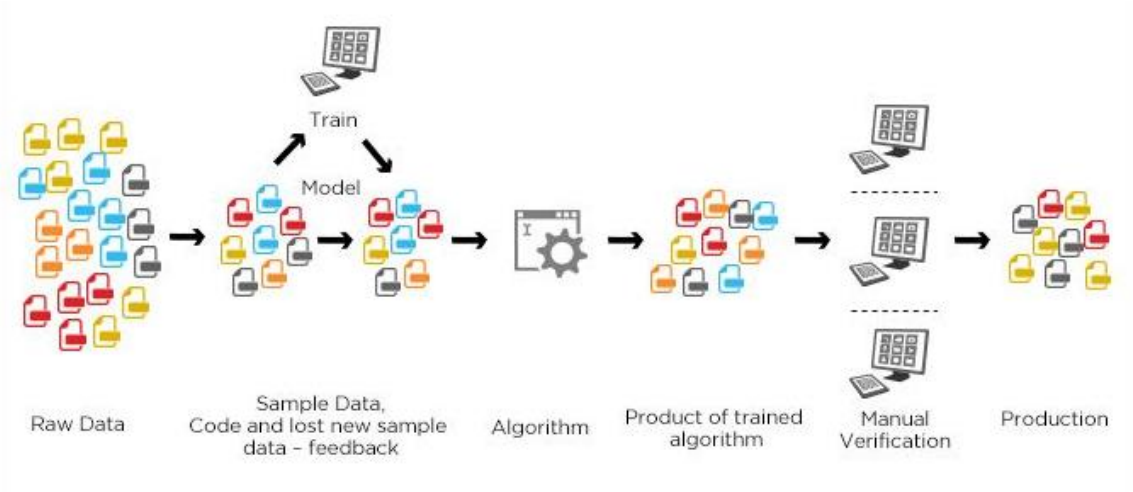
Jedna z nejdůležitějších částí strojového učení je pochopení dat, s kterými pracujeme a jejich propojení s cílem našeho problému, který řešíme. Je nezbytné pochopit, jak fungují naše data před tím, než začneme vytvářet model. Algoritmy jsou rozdílné a každý se používá na jiný typ problému (17, str.4).

Strojové učení můžeme rozdělit na tři části (9):



Obrázek 4: Typy strojového učení
(Zdroj: 9)

1. Učení s učitelem (supervised learning) (9).
2. Učení bez učitele (unsupervised learning) (9).
3. Reinforcement learning (9).



Obrázek 5: Průběh strojového učení
(Zdroj: 9)

1.5.1 Supervised learning (učení s učitelem)

V učení s učitelem vychází algoritmus z toho, že má trénovací a testovací data. Data máme rozdělena na vstupní a výstupní, obě tyto skupiny dat známe. Vstupní data přiřadíme proměnné X spolu s odpovídajícími výstupy, které značíme Y. Algoritmus se učí porovnáváním svých skutečných hodnot s výstupy, a to slouží k nalezení chyb. Poté model odpovídajícím způsobem upraví. Data rozdělí na trénovací a testovací. Procento testovacích a trénovacích dat si nastavujeme sami, přičemž by vždy mělo být větší procento dat na datech trénovacích. Trénovací data se používají pro zaškolení našeho algoritmu a na testovacích datech testuje trénovací algoritmus (9).
Učení s učitelem se využívá k predikci dat. Díky historickým datům dokáže predikovat budoucí události (9).

1.5.1.1 Regression (regrese)

Regrese je forma prediktivního modelu, která má vztah mezi závislou proměnnou (výstupem) a nezávislou proměnnou (vstupem). Tato technika je vhodná pro předpověď počasí, predikce ceny domu a další (9).

Linear Regression

Lineární regrese je základní a nejjednodušší algoritmus pro strojové učení. Predikujeme skóre z jedné proměnné z hodnocení od druhé proměnné. Proměnná, kterou předpovídáme se vždy nazývá Y a proměnná na, které se zakládají naše předpovědi se označuje jako X (9).

Multi linear Regression

Multi lineární regrese je nejčastější forma lineární regrese. Opět se používá k predikci a smyslem je, že máme vztah mezi jednou závislou proměnnou a několika nezávislýma proměnnýma (9).

Polynomial Regression

Další forma regrese, ve které je nezávislá proměnná větší než jedna. Nejvhodnější přímkou tady není přímka, ale křivka (9).

Support Vector Regression

Tento typ regrese nemusí být použitý pouze na problémy, které mají vztah s regresí, ale také na klasifikační problémy (9).

Bayesian Regression

Bayesova regrese umožňuje pracovat i se špatnými daty. Dokáže potlačit přebytečný šum, tak aby se zbytek dat mohl použít bez něj. Nejdůležitější je, že Bayesova regrese dokáže říct, která data jsou pro ni vhodná a která jsou nejistá (9).

Decision Tree Regression

Tato forma rozpadává data na menší části a rozhodovací strom ve stejnou dobu vyvíjí. Výsledkem je strom s rozhodovacími uzly (9).

1.5.1.2 Classification (klasifikace)

Klasifikace slouží k predikci dat, která nejsou ve spojité formě. Výstupy těchto informací nejsou vždy spojité a lineární. V této technice se algoritmus učí ze zadaných dat a poté na testovacích datech pozoruje nové chování. Opět jsou data tedy rozdělena na trénovací a testovací. Jeden z klasifikačních problémů může být kontrola, zda e-mail má spadnout do složky spamu nebo ne. Určuje se to na základě vycvičením algoritmu na různá nevyžádaná slova v e-mailech (9).

Logistic Regression/Classification

Měří závislost mezi závislou proměnnou, která je v určité kategorii a jednou nebo více nezávislými proměnnými pomocí logistické nebo sigmoidní funkce. Obecně se tato regrese používá tam, kde závislá proměnná může nabývat pouze dvou hodnot, například „Ano/Ne“ nebo „Chytrý/Hloupý“ (9).

K-Nearest Neighbours

KNN algoritmus je nejpoužívanější algoritmus v klasifikaci. Algoritmus hledá vždy nejbližší skupinu (9).

Support Vector Machines

Používá lineární klasifikátor do dvou tříd, ve kterých diskriminační klasifikátor definuje rozdělení nadroviny. Trénovací data leží v opačných poloprostorech.

Nadrovinou se v geometrii rozumí pro daný prostor dimenze n, jakýkoliv jeho podprostor dimenze $n-1$. V rovině je tedy nadrovinou každá přímka a v třírozměrném prostoru je nadrovinou každá rovina. V eukleidovském prostoru platí, že nadrovina prostor dělí na dva poloprostory (9).

Decision Tree Clasification

Definuje model ve stromové struktuře. Rozděluje data na menší podmnožiny. Konečným výsledkem je strom s rozhodovacími uzly. Rozhodovací uzel má dvě nebo více větví. Reprezentuje klasifikaci nebo rozhodnutí. První rozhodovací uzel, který odpovídá nejlepšímu predikátoru se nazývá kořenový uzel. Rozhodovací stromy mohou zpracovávat jak kategorická, tak numerická data (9).

Random Forest Classification

Je to soubor rozhodovacích stromů, většinou je algoritmus trénovaný metodou „bagging“, což je kombinace učebních modelů (9).

1.5.2 Unsupervised learning (učení bez učitele)

Druhý typ strojového učení se nazývá učení bez učitele, ve kterém neoznačená data jsou použita pro trénování algoritmu, což znamená, že používají stejná data, která nemají historický popis. Účelem je prozkoumat data a najít nějakou strukturu. Při učení bez učitele jsou data neoznačena a použijí se rovnou v algoritmu bez předběžného zpracování dat, kdy algoritmus nemůže rozdělit data na trénovací a testovací. Algoritmus podle datových segmentů vytvoří shluky dat s novým označením. Tato technika slouží dobře na transakčních datech. Dokáže identifikovat segmenty zákazníků s podobnými atributy, s nimiž pak lze v marketingových kampaních zacházet podobně. Nebo může najít hlavní vlastnosti, které oddělují zákazníky od sebe. Také se tato technika používá k segmentaci textových témat, doporučování položek a identifikaci odlehlých dat (9).

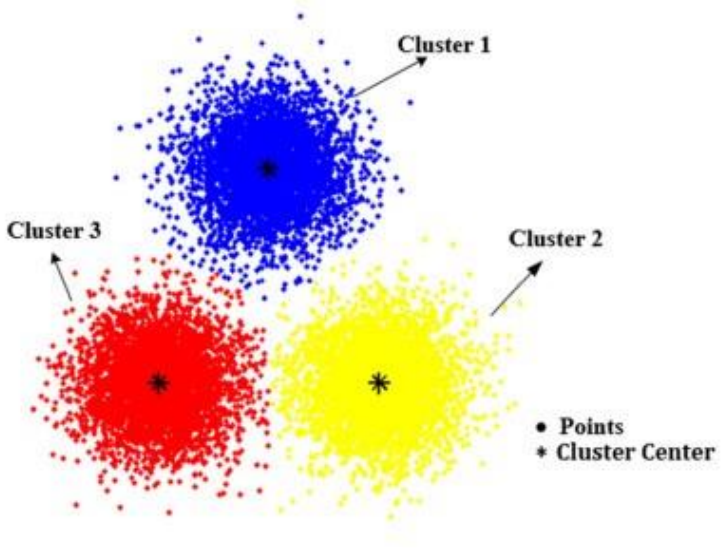
1.5.2.1 Clustering (shlukování)

Je to proces spojování podobných hodnot k sobě a poté spojená data využije pro vytváření shluků. Cílem této techniky je najít podobná data a seskupit je dohromady a zjistit, do které skupiny by měly nová data patřit (9).

K-Means Clustering

Je to jedna z technik shlukování, ve které jsou podobná data seskupena do shluků. Algoritmus se snaží najít lokální maxima v každé iteraci. Začíná se s K jako vstupem, který ukazuje kolik skupin chceme vidět. S použitím Euklidovské vzdálenosti se

vypočítá vzdálenost mezi daty a centrem shluku, a přiřadí data k bodu shluku, který je mu blízký. Přepočítává středy shluků jako průměr datových bodů, které jsou k němu připojené. Opakuje to, dokud nenastanou žádné další změny (9).



Obrázek 6: Clustering

(Zdroj: 9)

Hierarchical Clustering

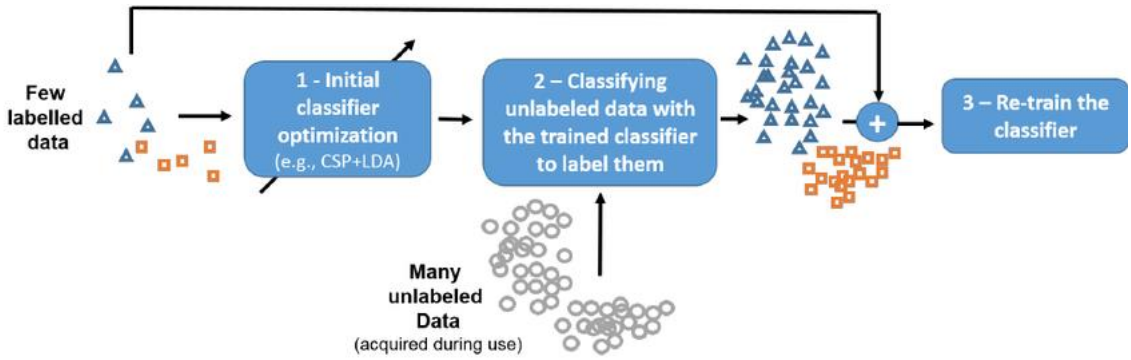
Tato technika opět spojuje podobná data do shluků. Vytváří hierarchii shluků. Začíná se všemi body, které jsou přiřazeny ke své vlastní skupině. Poté jsou dvě nejbližší skupiny sloučeny do stejného shluku. Algoritmus se ukončí, když zbyde pouze jeden shluk (9).

1.5.3 Reinforcment Learning

Reinforcment learning je třetí typ strojového učení, ve kterém nejsou zadaná žádná data jako vstup namísto toho musí algoritmus zajistit situaci sám. Používá se nejčastěji pro roboty, hraní a navigaci. Algoritmus odhalí pomocí metody pokus/omyl, které akce přináší největší odměny. Má tři složky, agenta, který je popsán jako učitel nebo tvůrce rozhodnutí. Prostředí, které popisuje vše, s čím agent spolupracuje a akce, co popisuje, co agent může dělat (9).

1.5.4 Semi-Supervised Learning

Je to hybridní řešení spojující učení s učitelem a bez učitele. Používá se pro stejné účely jako učení s učitelem, kde jsou neoznačená i označená data použita pro trénování. Tento typ učení lze použít s metodami jako jsou klasifikace, regrese a predikce. Tato technika je užitečná z několika důvodů. Za prvé proces značení velkého množství dat pro učení s učitelem je často neúměrně časově náročný. A moc velké označení může modelu předepsat lidské předpojatosti. To znamená, že zahrnutí velkého množství neoznačených dat do trénovacího procesu zlepšuje přesnost konečného modelu a zároveň snižuje čas i náklady na jeho vybudování (9).



Obrázek 7: Semi-Supervised Learing

(Zdroj: 9)

1.6 Python a Machine learning

Python začíná být hodně používaný pro datovou vědu. Je to dáno kombinací síly programovacího jazyka a jednoduchým používáním. Python má také knihovny pro načítání a vizualizaci dat, statistiku a další (17, str. 5).

Scikit-learn je velmi oblíbený nástroj a knihovna v Pythonu pro strojové učení. Závisí na dalších balíčcích v Pythonu. Pro práci s těmito balíčky používám Python Anaconda, který obsahuje všechny potřebné balíčky jako NumPy, SciPy, matplotlib, pandas a scikit-learn (17, str. 6).

Knihovna NumPy se používá pro využití multidimenzionálních polí, matematických funkcí jako jsou operace lineární algebry a využití generátoru pseudonáhodných čísel (17, str. 7).

Matplotlib slouží primárně na vykreslení grafů v Python (17, str.9).

Knihovna pandas slouží pro analýzu dat. Data jsou strukturovaná stejně jako v SQL databázích, souborech .csv nebo v Excelu. Tyto formáty dat jdou také pomocí knihovny pandas přímo do Pythonu nahrát. Pandas pracuje s datovým typem DataFrame, který znázorňuje tabulku (17, str. 10).

```
import pandas as pd

# create a simple dataset of people
data = {'Name': ["John", "Anna", "Peter", "Linda"],
        'Location' : ["New York", "Paris", "Berlin", "London"],
        'Age' : [24, 13, 53, 33]
       }

data_pandas = pd.DataFrame(data)
# IPython.display allows "pretty printing" of dataframes
```

	Age	Location	Name
0	24	New York	John
1	13	Paris	Anna
2	53	Berlin	Peter
3	33	London	Linda

Obrázek 8: Python, knihovna pandas

(Zdroj: 17, str. 10)

Na příkladu lineární regrese ukážu použití programovacího jazyka Pythonu. Jedná se o nejjednodušší metodu pro regresi. Importujeme si lineární model do Pythonu. Na model použijeme metodu fit(), která najde vztah mezi jednotlivými hodnotami (17, str. 49).

```
from sklearn.linear_model import LinearRegression
X, y = mglearn.datasets.make_wave(n_samples=60)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

lr = LinearRegression().fit(X_train, y_train)

print("lr.coef_:", lr.coef_)
print("lr.intercept_:", lr.intercept_)
```

Obrázek 9: Python, lineární regrese

(Zdroj: 17, str. 49)

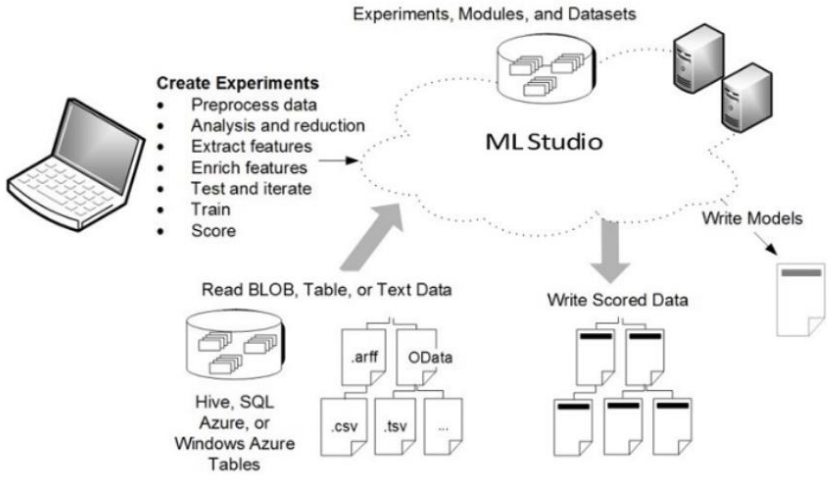
Pomocí parametru zavoláme koeficienty, které jsou uložené v atributu coef_ a atribut intercept, který je uložen v intercept_ (17, str. 49).

1.7 Azure Machine Learning Studio

Azure Machine Learning Studio je cloudová služba, která slouží pro budování, testování a implementaci prediktivního modelu založeného nad našimi daty. Využívá „drag-and-drop“ nástroj pro budování modelů. Je interaktivní a je možné jej upravovat odkudkoliv chceme. Rovněž lze konvertovat trénovací experiment na prediktivní experiment a publikovat jako webovou službu (18).

Pravidla pro práci s experimenty:

- Experiment musí mít alespoň jednu datovou sadu a jeden modul (18).
- Datová sada může být připojena pouze k modulu (18).
- Všechny vstupní porty pro moduly musí mít nějaké připojení k datovým sadám (18).
- Všechny parametry pro jednotlivé moduly musí být nastavené (18).

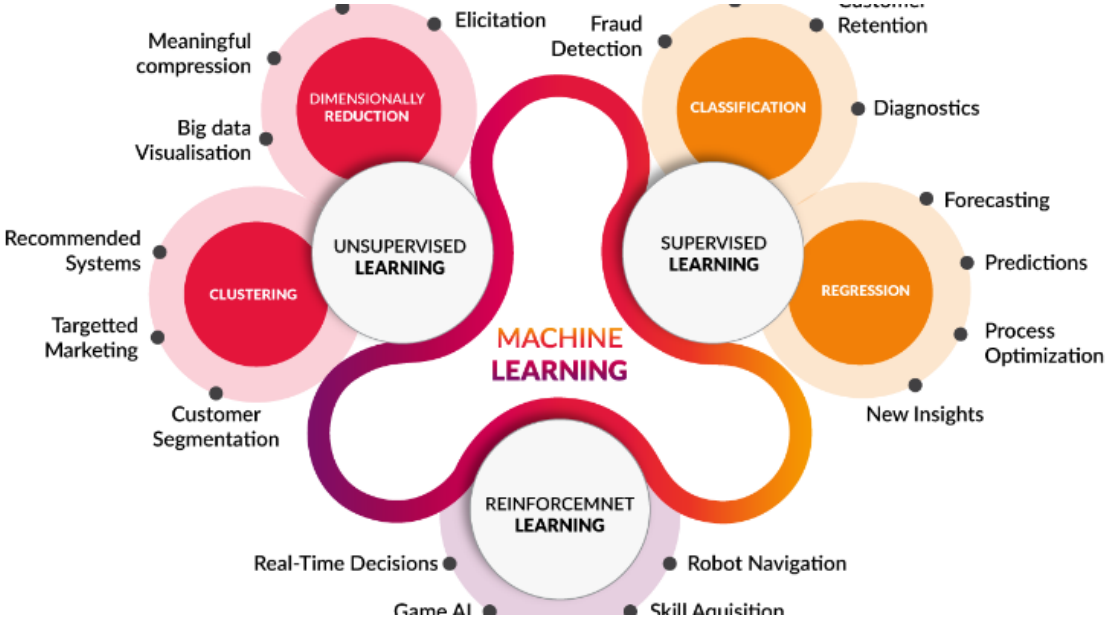


Obrázek 10: Microsoft Azure Machine Learning Studio

(Zdroj: 18)

1.8 Využití Machine learning

Strojové učení se dá využít v mnoha odvětvích např. zdravotnictví, technologiích, financích, bezpečnosti a další. Na obrázku níže je znázorněno, kde se používá, jaká technika a jaký typ strojového učení (9).



Obrázek 11: Využití strojového učení

(Zdroj: 9)

1.9 Analytické nástroje

1.9.1 SLEPT analýza

Analýza slouží k odhalení vnějších faktorů firmy. Podle prvních písmen názvu analýzy se dělí na tyto části (10).

- **Sociální oblast** – životní úroveň, demografické faktory a struktura populace, vzdělání, míra nezaměstnanosti (10).
- **Legislativní oblast** – právní normy týkající se společnosti (10).
- **Ekonomická oblast** – daňové zatížení, fáze hospodářského cyklu, měnová politika (10).
- **Politická oblast** – vliv politických stran, stabilita politického systému, války, demonstrace (10).

- **Technologická oblast** – technologická vyspělost, rychlost vývoje, podpora ve výzkumu (10).

1.9.2 McKinseyho model 7S

Je to analytická technika, která slouží k hodnocení kritických faktorů v organizaci. Má sedm prvků hodnocení (11).

- **Skupina** – definování společenství lidí (11).
- **Strategie** – definování cílů, vize v organizaci (11).
- **Sdílené hodnoty** – firemní kultura v organizaci, firemní poslání (11).
- **Schopnosti** – schopnosti a dovednosti zaměstnanců ve firmě (11).
- **Styl** – styl jednání v organizaci, komunikace mezi nadřízenými a podřízenými (11).
- **Struktura** – organizační struktura firmy (11).
- **Systémy** – použití informačních systémů ve firmě, komunikačních nástrojů, metody, procesy a postupy ve firmě (11).

1.9.3 Porterova analýza pěti sil

Tento model pracuje s pěti prvky, které se týkají organizace. Analyzuje odvětví a jeho konkurenční situaci i s riziky, která mohou nastat (12).

- **Stávající konkurence** – schopnost ovlivnit cenu a nabízené množství produktu (12).
- **Potencionální konkurence** – potencionální hrozba vstupu nové konkurence na trh a ovlivnění ceny či množství produktu (12).
- **Dodavatelé** (12)
- **Odběratelé** (12)
- **Substituty** – produkt, který lze částečně nahradit daný produkt (12).

1.9.4 SWOT analýza

SWOT analýza se používá pro zhodnocení vnitřních a vnějších faktorů, které ovlivňují úspěšnost organizace. Je to analytická technika, která se nejčastěji používá v rámci

strategického řízení a marketingu. SWOT analýzu lze rozdělit na čtyři části, které vycházejí z počátečních písmen (13).

- Strengths – silné stránky (13).
- Weaknesses – slabé stránky (13).
- Opportunities – příležitosti (13).
- Threats – hrozby (13).

Před vytvořením SWOT analýzy je vhodné použít další metody, které pomohou při jejím finálním vytvořením. Pro vnitřní faktory se dá využít finanční analýza organizace, McKinseyho model 7S nebo analýza produktového portfolia. Pro vnější faktory je možné využít SLEPT analýzu anebo Porterovu analýzu pěti sil (13).

2 ANALÝZA SOUČASNÉHO STAVU

V této kapitole se budu zabývat analýzou společnosti Solitea Česká republika a.s.

2.1 Analýza firmy Solitea Česká republika a.s.

Solitea Česká republika a.s. je produktově orientovaná firma, nabízející systémy pro řízení ekonomických procesů pro živnostníky nebo také malé a střední podniky.

Své počátky má v roce 1990, kdy Martin Cígler založil společnost pod názvem Cígler Software a začali v něm programovat první účetní software Money verze 1. V následujících pěti letech byl tento program jedničkou mezi účetními systémy v Česku a od roku 1993 i na Slovensku (14).

V roce 2003 vypustili do světa účetní software Money S3, který se rychle zařadil mezi nejoblíbenější systémy pro malé firmy a živnostníky – jen v Česku ho používá více než dvacet tisíc uživatelů (14).

Protože firmy, které používaly Money S3 se začaly také rozrůstat, tak jim toto řešení přestávalo stačit. Tím pádem se společnost pustila do vývoje ERP systému Money S5 (14).

Další milník v životě firmy nastal, když spustili cloudovou aplikaci iDoklad, která běží na platformě Windows Azure (14).

Firma se neustále rozrůstá a vstupuje do holdingu Solitea, díky kterému si otevřeli bránu do Evropy. Aby byli v rámci holdingu čitelnější přejmenovali se v roce 2017 z Cígler Software na Solitea Česká republika a.s. (14).

Obrázek 12: Logo firmy Solitea Česká republika a.s.

(Zdroj: 14)

2.2 Popis produktů

money S3

Program, který slouží pro živnostníky a společnosti s ručením omezeným. Největší obrat firmě dělá právě tento produkt. Využívá ho přes 16 tisíc malých a středních firem (15).

informační systém money S4

Je to ERP systém, který vychází z Money S5 a prodává se jako krabicové řešení. Nabízí všechny vlastnosti robustních systémů, avšak s rychlým nasazením a příznivou cenou (15).

informační systém money S5

ERP systém, který může být přizpůsobený každému zákazníkovi na míru. O jeho implementaci se starají konzultanti, kterým před nasazením popíše zákazník své představy o využití produktu (15).

iDoklad

Online fakturační systém, který využívá 200 tisíc živnostníků. Umožňuje daňové přiznání paušálem i podle skutečných výdajů nebo přímé propojení s externí účetní. Disponuje také přehledy o vystavených i přijatých fakturách, prodejkách, přijatých platbách či úhradách a řadou dalších funkcí (15).

prodejna SQL

Pokladní systém pro malé i střední prodejny. Má možnost centrální správy dat a vzdálenému monitoringu, díky tomu jej využijí i obchodní řetězce (15).

Profi Účtenka

Bezplatná aplikace Profi Účtenka umožňuje odkudkoliv vystavovat a odesílat účtenky v souladu s EET. V desktopové i mobilní verzi aplikace je možné spravovat sklady, ceníky či ji propojit s iDokladem. Dokáže fungovat v offline režimu (15).

2.3 SLEPT Analýza

Sociální faktory

Solitea Česká republika má hlavní sídlo v Brně, kde se nachází management firmy a většina zaměstnanců. Další pobočku mají v Praze, kde se nachází konzultanti a obchodníci jednotlivých produktů. Brnu se ne nadarmo přezdívá studentské město. Velké množství univerzit je pro společnost vhodné k tomu, nabídnout studentům praxi přímo při studiu nebo bezprostředně po něm. Průměrný věk ve společnosti je 35 let.

Legislativní faktory

Jako firma vyvíjející účetní software ji legislativní faktory velmi ovlivňují. Dotýká se jí zákon o účetnictví, daňové zákony, zákon o evidenci tržeb. Tyto zákony, ale nejsou jen zájmem účetních ve firmě. Jelikož firma prodává účetní software, musí tyto zákony znát i testeři jednotlivých produktů nebo konzultanti, kteří pravidelně jakoukoliv změnu zákona prezentují zákazníkům. Další legislativa, která se dotýká společnosti je zákon o GDPR, který přišel v platnost v roce 2018 a zabývá se ochranou osobních údajů.

Ekonomické faktory

Solitea Česká republika je akciová společnost. Největším zdrojem financí společnosti je produkt Money S3, který poskytují od samého začátku.

Politické faktory

Společnost ovlivní i některá politická rozhodnutí vlády. Aktuálně například nový zákon o dani z příjmů od ministryně financí. V minulých letech se ji například dotýkalo

rozhodnutí od premiéra a bývalého ministra financí ohledně evidence tržeb. Tyto změny je nutné zapracovat do produktů společnosti.

Technologické faktory

Firma využívá k vývoji nejnovější technologie od Microsoftu, s kterým je Gold Partnerem. Vyvíjí na platformě .NET, používají MSSQL server. K tvorbě reportů využívají službu MS Power BI a MS Integration Services. V oblasti technologií firma svým zaměstnancům dává možnost se realizovat a vzdělávat. Díky tomu se kvalita vývoje stále posouvá.

2.4 Analýza 7S organizace

Strategie

Firma úspěšně usnadňuje život malým živnostníkům, ale i korporacím nebo státní správě. Cílí jak na menší podnikatelé, tak na střední až větší firmy, kde nabízí Money S4 a S5. Nebojí se investovat do vývoje a díky tomu mohou zákazníkům nabídnout už teď to, o čem ostatní jen sní.

Systémy

Organizace používá jako hlavní systém svoje ERP řešení, Money S5. Docházku, plány dovolené evidují pomocí systému Vema.

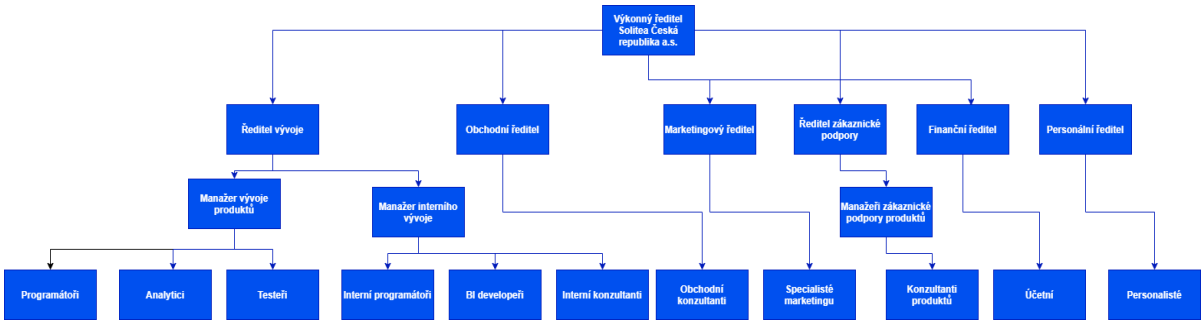
Ke komunikaci slouží především Microsoft Teams, MS Outlook a v krajních případech se ještě využívá Skype for Business. Pro projektové řízení je využíván Azure DevOps, ve kterém probíhá i plánování úkolů.

Sdílené hodnoty

Ve firmě panuje neformální atmosféra a společnost sdílí několik hodnot. Mezi ně patří **dobrá nálada**, protože v práci lidé tráví velkou část dne a je důležité, aby do práce chodili rádi. V týmech se snaží udržovat pozitivní energie. Další je **podpora**, a to nejen pro zákazníky, ale také mezi kolegy, kdy se vzájemně snaží podporovat. Zakládají si také na **otevřené komunikaci**, a to se týká nového nápadu nebo nějaké připomínky, nenechávají si věci pro sebe. Definují **odpovědnost** v rámci jednotlivých týmů i mezi týmy. Dodržují

slovo i domluvený termín. Ve firmě oceňují **inovativní** nápady, jejich dlouhodobým cílem je poskytovat inovativní produkty, které budou o krok před konkurencí. Jelikož každá firma stojí na zaměstnancích, nešetří se na jejich **rozvoji** a průběžném vzdělávání.

Struktura



Graf 1: Organizační struktura

(Zdroj: Vlastní zpracování)

Styl

Komunikace mezi podřízenými pracovníky a manažery je otevřená. Zaměstnanci mají možnost se vyjádřit a pokud dochází k nějakým neshodám, snaží se najít společnou cestu. Vývojové týmy jsou řízeny agilní metodou Scrum. Fungují ve dvoutýdenních sprintech a na jejich konci si pracovníci navzájem prezentují, co kdo za daný sprint zvládl a plánují společně dané úkoly. Jestli daný úkol může splnit více různých pracovníků, mají volbu rozhodnout se, kdo co bude chtít dělat.

Spolupracovníci

Firma zaměstnává jak zkušené pracovníky, tak se také nebojí zaměstnat studenty nebo juniornější pracovníky a poskytnout jim prostor pro vzdělání a zlepšení. Snaží se, aby toto v týmech bylo vyvážené a aby juniornější pracovníci měli možnost kdykoliv a s čímkoliv obrátit se na někoho zkušenějšího.

Schopnosti

Schopnosti zaměstnanců jsou ve firmě různorodé. Rozdělila bych je na IT pozice, do kterých spadají programátoři, testeři, analytici, kteří mají převážné dobré logické myšlení a umí data rozumně posoudit a vyhodnotit je. Pak jsou tu pozice, které se zabývají vztahy

mezi lidmi, což je třeba personální oddělení. Anebo vztahy se zákazníky a celková péče o ně.

2.5 Porterova analýza pěti sil

Konkurence

Konkurence společnosti by se mohla rozdělit na konkurenci jednotlivých produktů. Firma vytváří rozmanitý účetní software a každý produkt má svou vlastní konkurenci.

Začnu produktem Money S3, který je nejstarší. Největším konkurent je produkt POHODA, který cílí stejně jako Money S3 na podnikatelé a menší firmy. Money S4/S5 spadá do kategorie ERP systémů, kde je konkurence opravdu vysoká. Můžu zmínit K2 systém, ABRA systém anebo SAP, i když ten už cílí na mnohem větší zákazníky.

Pro mě je ale nejdůležitější produkt iDoklad, kterého bude nejvíce týkat tato práce. Konkurence iDokladu je online správa faktur Vyfakturuj od firmy Redbit s.r.o. Oproti iDokladu má výhodu, že levnější tarif Hobby+, který je za 75 Kč/měsíc nabízí neomezené množství kontaktů a faktur. U iDokladu je toto možné až při tarifu „Základní“, který má ale cenu 145 Kč/měsíc. Firma Redbit s.r.o. nabízí také nejvyšší tarif za cenu nižší o 100 Kč/měsíc než iDoklad. Další konkurent je online fakturace Fakturoid od firmy Fakturoid s.r.o. Ceny jsou podobné jako u iDokladu, ale nejvyšší tarif umožňuje připojení menšímu množství uživatelů než u iDokladu. Fakturoid má nejdražší tarif zaměřený na větší firmy.

Potencionální konkurence

Na tomto trhu je možnost vstupu nové konkurence. Firem, které se zabývají informačními nebo účetními softwary je spousta, takže přijít s cloudovou aplikací na fakturace pro ně není nereálné. Produkt iDoklad je podle NPS analýzy nejlépe hodnocený produkt ve firmě (nad 9 bodů), takže zájmem firmy je udržet si tento stav nejen uvnitř firmy, ale podpořit ho i na celém trhu. Solitea staví na tom, že má velmi známé jméno v České a Slovenské republice (hlavně pod bývalým názvem CÍGLER software s.r.o.) a její hlavní činností podnikání je účetní software.

Dodavatelé

Největším dodavatelem společnosti je Microsoft. Jeho produkty využívá společnost denně při své práci. Vývojáři a manažeři využívají Azure DevOps pro plánování práce, portál Azure je taktéž přístupný pro vývojáře ve firmě, kteří jej využívají pro svou činnost. Každý zaměstnanec používá ke své práci také Office 365.

Odběratelé

Tím, že trh nabízí velké množství podobných produktů, vyjednávací síla odběratelů je silná. Společnost má velkou škálu produktů, jak pro větší firmy, tak i pro menší firmy a živnostníky.

Substituty

Produkty Money S4 a Money S5 jsou snadno zaměnitelné, jelikož S4 je „krabicové ERP řešení“ a S5 je systém přímo na míru. Další produkt iDoklad je jediný, který běží na cloudu. Už kvůli tomu je jeho substituce s ostatními produkty problematická. Také je to specifický produkt, který se zaměřuje převážně na fakturování a slouží tedy menším živnostníkům. Nejblíže k iDokladu by pro zákazníky mohl být vhodný produkt Money S3.

2.6 SWOT analýza firmy

Tabulka 1: SWOT analýza firmy

(Zdroj: Vlastní zpracování)

Silné stránky	Slabé stránky
• inovativní produkty • péče o zákazníky • přátelská atmosféra ve firmě • 29 let působení na trhu • rozvoj zaměstnanců • známé a hojně používané produkty • vlastní blog s novinkami money.cz • velké množství dat pro tvorbu analýz • rozhodování manažerů je podloženo daty	• na trhu ERP systémů silně konkurenční odvětví • nedostatek pracovníků na konzultantských pozicích • špatné plánování kapacit jednotlivých týmů
Příležitosti	**Hrozby**
• nové produkty zaměřené na cloud technologii • zmodernizování ERP systému Money S4/S5 • lepší využití velkého množství dat • rozvoj a udržení si dobrého hodnocení produktu iDoklad na trhu	• odchod kvalitních zaměstnanců • vznik nové konkurence • příchod ekonomické krize • různé legislativní změny v rámci účetnictví

2.7 Produkt iDoklad

V této diplomové práci se budu zabývat hlavně produktem iDoklad. Z toho důvodu jsem se rozhodla provést krátkou analýzu i na něm.

V roce 2011 spustil tehdy ještě CÍGLER software čistě cloudovou aplikaci na platformě Windows Azure. Díky dobrému načasování oslovili podnikatele, kteří doposud používali papír a tužku anebo Excel (16).

iDoklad je možné mít ve čtyřech různých tarifech. Tarif Zdarma, Základní, Oblíbený a Prémiový (16).

Obrázek 13: Tarify iDoklad

(Zdroj: 16)

Kromě těchto základních funkcí obsahuje iDoklad i další možnosti. Například položky v ceníku, napojení na účetní software, vlastní mobilní aplikaci. Nejvyšší tarif obsahuje až 9 uživatelů, kteří mohou iDoklad obsluhovat. Na vyzkoušení je zde využívání všech funkcí iDokladu zdarma na 2 měsíce, aby uživatel poznal všechny výhody a mohl se rozhodnout, zda bude využívat placenou verzi produktu (16).

Co se týče zákaznické spokojenosti v letech 2018 a 2019 tento produkt vyšel jako nejlépe hodnocený produkt ve firmě v rámci NPS analýzy. I když společnosti nepřináší tento produkt nejvyšší zisk, dokáže si díky němu budovat velkou a spolehlivou klientelu a případně je seznámit s dalšími produkty Solitei.

Oproti ostatním produktům má tento výhodu v tom, že shromažďuje velké množství dat od svých zákazníků. Kromě klasických informací jako objem fakturace uživatelů, prodeje minulých let, tak obsahuje také logovací záznamy, díky kterým můžeme zjistit, kdy se zákazník k webové aplikaci připojil nebo například jaké funkcionality v iDokladu nejčastěji využívá.

2.8 Databáze

Firma má několik databází. Pro účely Business Intelligence a analýz se používá datový sklad, ze kterého se data používají v reportech. Ve většině případech se data do datového skladu tahají z ostrých databází.

V ostré databázi se nachází data především z firemního ERP systému Money S5. Dále do ostré databáze putují data z API (Application Programming Interface), kterou využívá iDoklad a ProfiÚčtenka.

Na ostrém serveru jsou přístupy poskytovány přes účty. Programátoři, kteří spravují tuto databázi a potřebují ji ke každodenní práci využívají administrátorský účet. BI developeři, kteří pracují převážně v datovém skladu mají vytvořený na ostré databázi speciální účet pro přístupy jen k databázím, které aktuálně využívají. Je podstatné nedávat přístupy všem IT zaměstnancům ve firmě, ale opravdu jen lidem, kteří potřebují ostrou databázi ke své denní pracovní rutině.

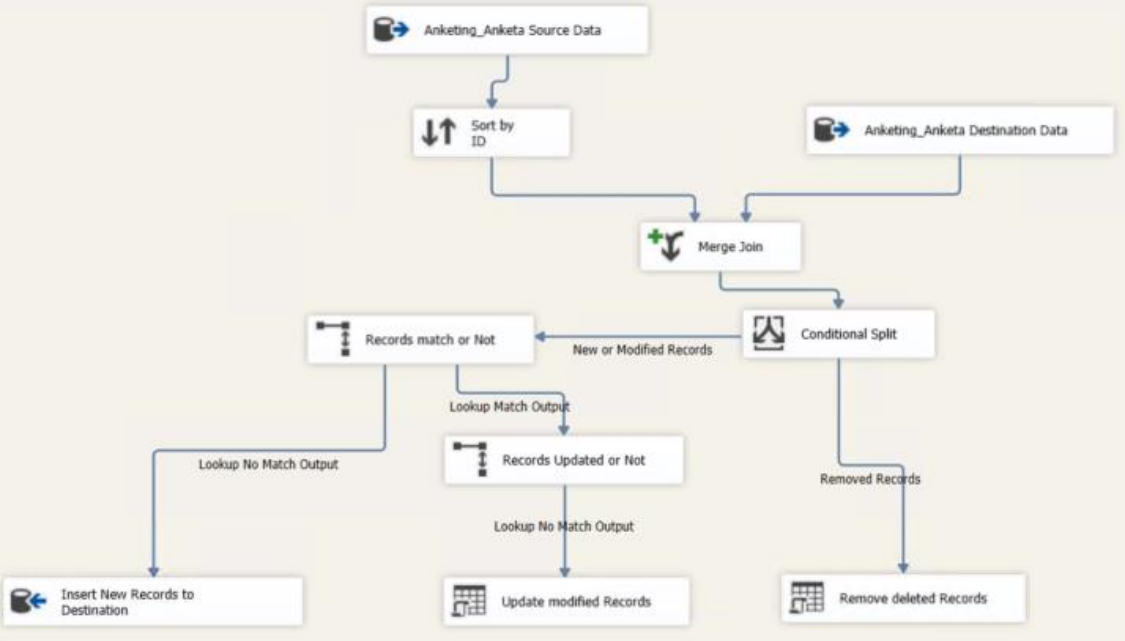
Ve své diplomové práci se budu převážně zabývat databázemi, které se nachází v datovém skladu.

Tahání dat do datového skladu

Pro přesunutí dat z ostré databáze používá firma nástroj Integration Services (SSIS), který je součástí balíčku Microsoft Server Data Tools.

Při prvním načtením se vezmou veškerá data z ostré databáze a nasypou se do námi vytvořené tabulky v SQL Management Studiu. Potom se už data načítají inkrementálně. Porovnávají se záznamy z tabulky z ostré databáze a ID, které už je v datovém skladu. Pokud se tedy objeví nový záznam, což se pozná podle ID, tak se vytvoří nový záznam.

Při změně záznamu se zkontroluje Datum, kdy byl záznam upraven a do datového skladu se nahraje ten novější. Pokud ID v ostré databázi není, z datového skladu se smažou.



Obrázek 14: Tahání dat z ostré databáze do datového skladu

(Zdroj: Vlastní zpracování)

3 VLASTNÍ NÁVRH ŘEŠENÍ

Díky analýze současného stavu jsem odhalila nedostatky, na které se ve firmě můžu zaměřit. Jako můj návrh poslouží zavedení strojového učení do firemního prostředí, které se bude provádět nad firemními daty. Tento návrh jsem si vybrala z toho důvodu, že firma má obrovské množství nevyužitých dat, hlavně právě z produktu iDoklad, kde zaznamenává mimo jiné logy, kdy se zákazník přihlásil a které funkce iDokladu využíval. Strojové učení může firmě přinést konkurenční převahu, jelikož blíže pozná svého zákazníka.

Před začátkem samotného návrhu řešení chci podotknout, že veškeré názvy tabulek a citlivé informace o zákaznících jsou pseudonymizované.

3.1 Plán projektu

Hlavní cíl projektu je začít využívat strojové učení ve firemním prostředí a díky vytvořeným modelům získat přesnější predikce do budoucna. V tomto projektu se budu zabývat dvěma zkušebními modely, do budoucna se počítá s použitím více modelů. Hlavní přínos je ulehčení práce manažerům ohledně strategického plánování vývoje firmy.

3.1.1 Projektový tým

Hlavní zodpovědnost za projekt bude mít projektový manažer, který má ve svém týmu jak programátory, tak Business Intelligence developery (v textu používám zkrácený název BI developer), kteří se tomu převážně budou věnovat. Po konzultaci s výkonným ředitelem ví, čeho chce dosáhnout a má znalosti ohledně datové problematiky, tudíž může tento projekt řídit. Činnosti projektu budou vykonávat BI developeři, ve spolupráci s programátory, kteří mají zase jiný pohled na věc.

3.1.2 Časová analýza

Metoda PERT

Pro tento projekt jsem se rozhodla použít metodu PERT, která se používá u složitějších projektů. Dobu trvání činnosti si zvolím optimistickou, pesimistickou a reálnou. Tento

projekt je ve firmě úplně nový. Při plánování budu vycházet ze zkušeností developerů a jejich datovými znalostmi. Tento projekt nebude vytvářet pouze jeden developer, ale dva, kteří spolu navzájem budou komunikovat a pomáhat si. Tím pádem při plánování byli přítomni dva BI developeři a projektový manažer a společně dávali návrhy na možné doby trvání.

Očekávaná doba trvání činnosti – neboli střední hodnota:

$$t_{ij} = \frac{a_{ij} + 4 * m_{ij} + b_{ij}}{6}$$

Směrodatná odchylka:

$$\sigma_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Rozptyl:

$$\sigma_{ij}^2 = \left(\frac{b_{ij} - a_{ij}}{6}\right)^2$$

a = optimistický odhad

b = pesimistický odhad

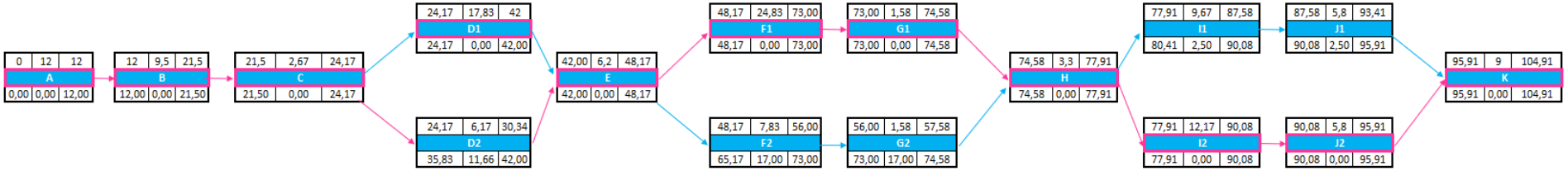
m = reálný odhad

Pro měření délky trvání jsem použila jednotku hodin, přičemž výslednou hodnotu přepočítám také na dny. Ve společnosti se plánuje při osmihodinové pracovní směně 6 hodin pracovního času. Tedy pokud bude nějaká činnost trvat 6 hodin, znamená to, že zabere celý pracovní den.

Tabulka 2: Časová analýza

(Zdroj: Vlastní zpracování)

Činnosti	Název činnosti	Navazující	aij	mij	bij	tij	oyij	σ2yij
A	Výběr a analýza vhodných dat	B	9	12	15	12,00	1,00	1,00
B	Pro jednotlivé predikce najít vhodné parametry, které ovlivní model	C	6	9	15	9,50	1,50	2,25
C	Konzultace s projektovým manažerem	D1, D2	2	2,5	4	2,67	0,33	0,11
D1	Čištění a úprava dat pro první model	E	14	18	21	17,83	1,17	1,36
D2	Čištění dat a úprava pro druhý model	E	4	6	9	6,17	0,83	0,69
E	Vybrání daného modelu	F1, F2	4	6	9	6,17	0,83	0,69
F1	Vytvoření prvního modelu	G1	21	24	32	24,83	1,83	3,36
F2	Vytvoření druhého modelu	G2	6	8	9	7,83	0,50	0,25
G1	Hodnocení daného modelu	H	1	1,5	2,5	1,58	0,25	0,06
G2	Hodnocení daného modelu	H	1	1,5	2,5	1,58	0,25	0,06
H	Konzultace s projektovým manažerem	I1, I2	2	3	6	3,33	0,67	0,44
I1	Predikce prvního modelu	J1	6	9	15	9,50	1,50	2,25
I2	Predikce a znázornění druhého modelu v Power BI	J2	9	12	16	12,17	1,17	1,36
J1	Obchodní manažer dle svých zkušeností zhodnotí model	K	3	6	8	5,83	0,83	0,69
J2	Marketingový manažer dle svých zkušeností zhodnotí model	K	3	6	8	5,83	0,83	0,69
K	Zavedení modelů na predikce do praxe		6	9	12	9,00	1,00	1,00



Graf 2: Grafické zpracování časové analýzy

(Zdroj: Vlastní zpracování)

V síťovém grafu je kritická cesta z činností **A-B-C-D_1-E-F_1-G_1-H-I_2-J_2-K.**

Díky výpočtu podle metody PERT jsem zjistila, že projekt zabere ve firmě 104,91 hodin čistého pracovního času, a tedy v přepočtu na den to bude **17,49 dnů.**

3.1.3 Matice odpovědnosti (RACI matice)

Matice odpovědnosti je metoda, která se používá při řízení projektů. Slouží pro rozdělení a přiřazení odpovědnosti jednotlivým členům týmům v projektech. Písmenka R A C I značí odpovědnost za daný úkol.

Tabulka 3: RACI matice – projektový tým

(Zdroj: Vlastní zpracování)

Činnosti	Název činnosti	Projektový manažer		BI developer		BI developer		Programátor
A	Výběr a analýza vhodných dat	A	C		R		R	C
B	Pro jednotlivé predikce najít vhodné parametry, které ovlivní model	A	C		R		R	C
C	Konzultace s projektovým manažerem	R	A		R		R	
D1	Čištění a úprava dat pro první model			R	A		C	C
D2	Čištění dat a úprava pro druhý model				C	R	A	C
E	Vybrání daného modelu	I		A	R		R	
F1	Vytvoření prvního modelu	I		R	A		C	
F2	Vytvoření druhého modelu	I			C	R	A	
G1	Hodnocení daného modelu	I		R	A		C	
G2	Hodnocení daného modelu	I			C	R	A	
H	Konzultace s projektovým manažerem	R	A		R		R	
I1	Predikce prvního modelu	I		R	A		C	
I2	Predikce a znázornění druhého modelu v Power BI	I			C	R	A	
J1	Obchodní manažer dle svých zkušeností zhodnotí model		A		R		R	
J2	Marketingový manažer dle svých zkušeností zhodnotí model		A		R		R	
K	Zavedení modelů na predikce do praxe	R	A		R		R	C

Tabulka 4: RACI matice – externí pracovníci

(Zdroj: Vlastní zpracování)

Činnosti	Název činnosti	Obchodní manažer	Marketingový manažer
A	Výběr a analýza vhodných dat	I	I
B	Pro jednotlivé predikce najít vhodné parametry, které ovlivní model		
C	Konzultace s projektovým manažerem		
D1	Čištění a úprava dat pro první model		
D2	Čištění dat a úprava pro druhý model		
E	Vybrání daného modelu		
F1	Vytvoření prvního modelu		
F2	Vytvoření druhého modelu		
G1	Hodnocení daného modelu		
G2	Hodnocení daného modelu		
H	Konzultace s projektovým manažerem		
I1	Predikce prvního modelu		
I2	Predikce a znázornění druhého modelu v Power BI		
J1	Obchodní manažer dle svých zkušeností zhodnotí model	R	
J2	Marketingový manažer dle svých zkušeností zhodnotí model		R
K	Zavedení modelů na predikce do praxe	I	I

3.1.4 Analýza rizik

Při plánování projektu je důležité analyzovat rizika a připravit opatření. Díky tomu zjistíme, zda projekt pro nás není příliš rizikový a můžeme ho zahájit.

Pro tuto analýzu jsem si vybrala skórovací metodu, která zahrnuje identifikaci rizika, ohodnocení a opatření pro snížení rizika.

V první řadě je potřebné nastavit si možnost výskytu tohoto rizika. Stupnice bude od 1 do 10, přičemž 10 je největší možnost výskytu a 1 nejmenší. Při dopadu používám stejnou stupnici jako při možnosti výskytu rizika. Celkovou hodnotu rizika si vypočítám vynásobením těchto dvou proměnných – možnosti výskytu a dopadu. Díky zavedení opatření se mi sníží možnost výskytu rizika, dopad zůstane stále stejný, ale cílem je snížit pravděpodobnost, že takové riziko vůbec nastane. Některá rizika mají výslednou hodnotu rizika nízkou, proto jsem se rozhodla podstoupit toto riziko projektu a nebudu pro ně dělat žádné opatření.

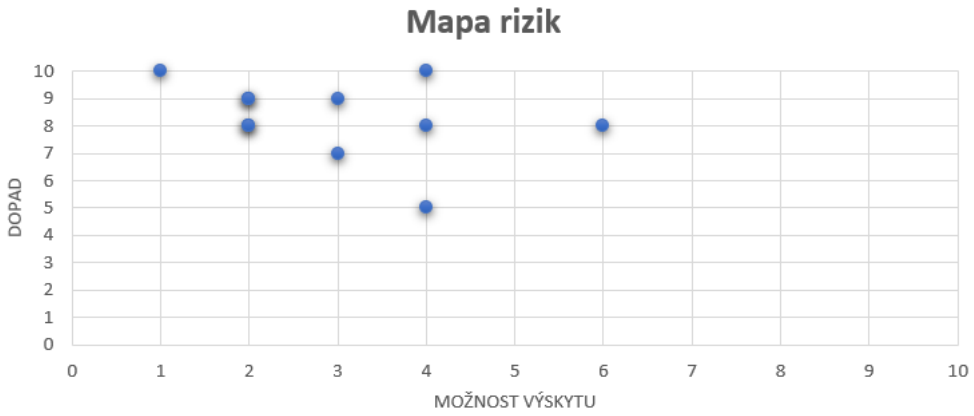
Tabulka 5: Analýza rizik

(Zdroj: Vlastní zpracování)

Riziko	MV	D	H	Opatření	NMV	ND	NH
Nedostatečné znalosti developerů o této technologii	3	7	21	Před samotným zahájením projektu, a i v jeho průběhu umožnit developerům školení na tuto technologii	1	7	7
Nevhodně vybraná prvotní data	3	9	27	Konzultace i s jinými programátory v týmu	2	9	18
Nevhodně vybrané proměnné, které ovlivňují model	2	9	18	Konzultace s manažery, kteří ze svých zkušeností budou vědět, jaké aspekty mají vliv na data	1	9	9
Nespokojenost manažerů s predikcemi	4	10	40	Do samotného procesu projektu zahrnout i konzultace s manažery	2	10	20
Špatně očištěná data (ponechány duplicity, prázdné hodnoty)	2	9	18	Kontrola druhým developerem, zda jsou data správně očištěná	1	9	9
Nevyužívání modelů do budoucna	6	8	48	Mít co nejpřesnější predikovaná data a tím ukázat manažerům přínos projektu pro ně	3	8	24
Management firmy přestane tento projekt podporovat	1	10	10	Podstoupení rizika	1	10	10
Špatně vybraný typ učení modelu	4	8	32	Dostatečně si prostudovat, který typ učení je pro danou predikci vhodný	2	8	16
Chybná implementace modelu	2	8	16	Vzájemná kontrola modelů developery	1	8	8
Prodloužení doby trvání projektu	4	5	20	Počítat s rezervou při plánování práce	2	5	10
Nedostatečné množství dat pro správné učení modelu	2	8	16	Podstoupení rizika	2	8	16

Mapa rizik

Mapa rizik slouží pro identifikaci, která rizika jsou pro nás nejdůležitější a měli bychom se jim věnovat. Jsou to především rizika v horní části grafu a také rizika v pravé polovině grafu. Tam se nachází rizika, která jsou pro projekt kritická a je důležité je patřičně snížit. Konkrétně v mém projektu je to riziko nevyužívání modelu do budoucna. Toto riziko je pro projekt významné hlavně proto, že mu předcházejí všechna ostatní rizika. Pokud bude model predikovat přesná a kvalitní data, neměl by být s využitelností projektu problém.



Graf 3: Mapa rizik

(Zdroj: Vlastní zpracování)

Zhodnocení opatření rizik

Po vytvoření daných opatření k jednotlivým rizikům je vidět, že je pro projekt stále nejrizikovější, že se nebude do budoucna využívat. Jak už jsem psala výše, toto závisí na více faktorech. Dále je třeba správně vybrat prvotní data, a také samotný typ učení pro model a do samotného procesu zapojovat manažery, aby byli s výsledkem spokojení. Riziko na nedostatečné znalosti ohledně technologie se mi podařilo snížit díky zaplánování školení na začátku projektu, i v jeho průběhu. Je velmi důležité, aby se developeři snažili, co nejvíce v této oblasti vzdělávat. Oproti rizikům bez opatření se výsledná hodnota snížila a nejvyšší nová hodnota dosahuje 24, původně nejvyšší hodnota byla 48.

3.2 Vytváření prediktivních modelů

Dle následujících bodů se budu řídit při vytváření prediktivních modelů:

1. **Definice problému –** Na začátek si musím definovat problém, který chci pomocí strojového učení řešit. Čeho díky predikcím dosáhnu?
2. **Výběr dat** – Tento bod se týká výběru vhodných dat, které můžu použít na tvorbu modelu. V této části také vyhledáváme spojitosti mezi daty, co která data ovlivňuje (19).
3. **Příprava dat** – Data před samotným vytvářením modelu je nutné upravit do podoby, ve které se mi s nimi bude lépe pracovat. To zahrnuje čištění dat, změna nebo úprava datového typu, napojení tabulek na sebe, doplnění dat. Tato část je v procesu strojového učení nejnáročnější a nejzdlouhavější (19).
4. **Výběr modelu** – Na základě dat, která máme a definovaného cíle vybereme vhodný model (19).
   Jednotlivé typy učení mám popsané v teoretické části.
5. **Trénink modelu** – V této části mnou vybraný algoritmus začne s učením se na datech. Před trénováním modelu si rozdělím data na trénovací a testovací, přičemž trénovacích by mělo být vždycky více. Ideální poměr je 80 % trénovacích dat a 20 % testovacích (19).
6. **Vyhodnocení** – V této části využiji druhou skupinu dat, díky které zjistím, zda vybraný model funguje správně (19).
7. **Ladění parametrů** – Může se stát, že model na první pokus nebude mít vysokou úspěšnost. Přesnost by se měla v ideálním případě pohybovat nad 90 %, to zaručuje minimální chybovost. Pokud těchto výsledků model nedosáhne je nutné vrátit se k druhému bodu a vybrat jiné souvislosti mezi daty (19).
8. **Predikce** – V této konečné části dostanu predikovaná data (19).

3.2.1 První prediktivní model

3.2.1.1 Definice problému

Služba iDoklad nabízí čtyři tarify: Zdarma, Základní, Oblíbený a Prémiový, přičemž první je zdarma a ostatní jsou placené. Jak už jsem psala výše díky množství dat, které se dostávají do databáze přes API, dokážu posoudit, které funkcionality, jaký zákazník nejvíc využíval a v které době.

Pro účely mé práce se zaměřím pouze na placené tarify. Budu se zabývat přechodem ze Základního tarifu na tarif Oblíbený. Cílem predikce bude doba, za kterou zákazník přešel z nižšího tarifu na tarif vyšší.

Díky zjištění, kdy konkrétní zákazník by mohl přejít na vyšší tarif, může marketingové oddělení začít nabízet zákazníkovi výhodnější koupi vyššího předplatného než za normálních okolností. Taktéž uvidí závislosti mezi různými funkcionalitami.

Výhodou tohoto modelu je, že na stejném principu je možné ve firmě v budoucnu provést predikci na délku přechodu z jakéhokoliv nižšího tarifu na vyšší. V další části práce jsou popsané použité kódy a dotazy pro práci s daty Základního a Oblíbeného tarifu.

3.2.1.2 Výběr dat

V databázi firmy jsem si vyfiltrovala všechny zákazníky, kteří mají tarif Základní nebo Oblíbený. Další bod, na který se musím zaměřit je určit si parametry, které mi ovlivní přechod zákazníka na vyšší verzi. Tyto parametry si určím z funkcionalit iDokladu.

Určené parametry:

- Počet ceníkových položek, které za dobu na nižším tarifu vytvořil – zákazník si vytvoří vlastní ceník, který pak může používat při fakturaci.
- Počet příloh, které za dobu na nižším tarifu vytvořil – k vystaveným i přijatým fakturám přiloží přílohu v různých formátech.
- Počet uživatelů, které měl v době na nižším tarifu.
- Počet nastavených upomínkových e-mailů za dobu na nižším tarifu.
- Druh operačního systému na mobilním zařízení.

- Používal vlastní logo a barevné schéma: ano/ne.
- Používal newsletter: ano/ne.

Predikované parametry:

- Počet měsíců, kdy zákazník přešel z nižšího tarifu na vyšší.

3.2.1.3 Příprava dat

V této fázi se budu věnovat přípravě dat a jejímu čištění. Z firemního datového skladu jsem si vyselektovala všechny zákazníky, kteří měli tarif Základní nebo Oblíbený. Záznamů mi vyjelo 153 tisíc, ale při bližším zkoumání jsem zjistila, že v záznamech je evidováno každé předplatné, které zákazník měl.

	Agenda	datefrom	dateto	SubscriptionType	DaysBetween
1	2139	2019-04-01 10:32:43.033	2019-05-07 23:59:59.000	2	36
2	2139	2020-03-03 07:47:21.897	2020-04-03 23:59:59.000	1	31
3	2139	2017-09-25 18:21:56.610	2017-12-25 23:59:59.000	1	91
4	2139	2018-03-29 15:45:32.250	2018-05-28 23:59:59.000	2	60
5	2139	2018-02-22 15:22:06.853	2018-03-29 15:45:31.250	1	35
6	2139	2019-01-13 17:45:37.110	2019-04-01 10:32:42.033	1	78
7	2139	2019-12-08 10:04:45.670	2020-01-08 23:59:59.000	1	31

Obrázek 15: Ukázka části z databáze

(Zdroj: Vlastní zpracování)

Agenda = zákazník

DateFrom = datum, kdy si zákazník zakoupil tarif

DateTo = datum, kdy zákazník ukončil předplatné tarifu

SubscriptionType = typ tarifu, může nabývat hodnot:

- 0: Zdarma
- 1: Základní
- 2: Oblíbený
- 3: Prémiový

V této tabulce jde vidět, že jedna Agenda měla tarify Základní i Oblíbené. První tarif, který tento zákazník měl, byl Základní. V roce 2018 (Id 4 v tabulce) přešel na tarif

Oblíbený. Když si sečtu dny, kdy měl Základní tarif, výsledná hodnota bude 126. Mě ale zajímá celková doba, jak dlouho zákazníkovi trvalo, aby od začátku používání Základního tarifu přešel na vyšší tarif. Tuto hodnotu si spočítám odečtením atributu „Datefrom“ na řádku 4 s hodnotou „DateFrom“ na řádku 3. Výsledná hodnota je 185. V mé práci nebudu řešit, že zákazník byl na nižším tarifu pouze 126 dní, ale to za jak dlouhou dobu se dostal na vyšší tarif. Z této tabulky je patrné, že zákazník měl tento tarif pouze 60 dní a potom se k němu vrátil až v roce 2019, kdy opět začal na tarifu Základním (řádek č. 6) a pak v tom stejném roce zase přešel na tarif Oblíbený. Já se budu zaměřovat pouze na jeho prvotní rozhodnutí, že přejde na vyšší tarif. Ostatní případy už dále nebudu řešit. Tento případ je specifický, většina zákazníků, kteří si zakoupili tarif Oblíbený, už na něm zůstali. Ale je nutné počítat i s tím, že takové anomálie se v tabulce vyskytují a uvažovat s nimi při další filtraci.

V následujícím kroku si vytáhnu nejmenší datum u tarifu Základního i Oblíbeného a nejvyšší datum u těchto obou tarifů. Toto udělám pomocí následujícího příkazu v jazyku SQL.

```
SELECT Agenda, MIN(CONVERT (DATE,datefrom)) AS datefrom,MAX(CONVERT(DATE,dateto))
AS dateto, SubscriptionType

        FROM SubscriptionAgenda
        WHERE SubscriptionType = '1'
        GROUP BY agenda, SubscriptionType

        UNION

SELECT Agenda, MIN(CONVERT(DATE,datefrom)) AS datefrom, MAX(CONVERT(DATE,dateto))
AS dateto, SubscriptionType

        FROM SubscriptionAgenda
        WHERE SubscriptionType = '2'
        GROUP BY agenda, SubscriptionType
ORDER BY Agenda
```

Výsledek je následující:

	Agenda	datefrom	dateto	SubscriptionType
1	2139	2017-09-25	2020-04-03	1
2	2139	2018-03-29	2019-05-07	2

Obrázek 16: Ukázka z databáze

(Zdroj: Vlastní zpracování)

Z tohoto výsledku už je možné spočítat počet dní, které mě skutečně zajímají. V atributu „DateTo“ jde i přesto vidět, že Základní tarif byl poslední tarif od tohoto uživatele. Toto nemusím řešit.

Z mého SQL skriptu si odmažu podmínku na číslo Agendy a udělám toto řešení nad všemi ostatními daty. Záznamů mi vyjede 44 540. V dotazu není zajištěna filtrace zákazníků, kteří mají pouze Základní tarif.

Další krok, který je potřeba udělat je odfiltrovat zákazníky, kteří měli někdy oba tarify a najít pouze ty, kteří měli prvně tarif Základní a až potom přešli na tarif Oblíbený. Následně vypočítám počet dní mezi nimi. Tento problém jsem řešila v programovacím jazyku Python, který se v rámci strojového učení a práci s daty často používá.

```
import datetime

data = open("C:\\....\\Tarify.csv")

records = []
for x in data:
    records.append(x)

cleared_data = []
for record in records:
    attributes = y.split(",")
    cleared_data.append(attributes)


def tarif_decider(data):
    basic = []
    favorite = []

    for part in data:
        if "1" in part[3]:
            time_part = time_converter(part[1])
            part[1] = time_part
            basic.append(part)
        elif "2" in part[3]:
            time_part = time_converter(part[1])
            part[1] = time_part
            favorite.append(part)
    return basic, favorite
```

Vyfiltrovaná data z datového skladu jsem si nahrála do .csv souboru a ten jsem následně nahrála do Pythonu. Následně jsem iterovala získanými daty, které byly přidány do pole, tak aby jeden prvek v poli reprezentoval jeden záznam. Dále jsem iterovala přes vytvořený list (records) a rozdělila jsem data podle sloupců pomocí funkce Split.

Vytvořila jsem si list pro Základní tarif „basic“ a Oblíbený tarif „favorite“. Do listů jsem si vložila datum, který jsem si konvertovala na datový typ datetime. Toto využívám v mé funkci time_converter.

```
def time_converter(time):
    time_format = datetime.datetime.strptime(time, "%Y-%m-%d")

    return time_format

def get_object(basic, favorite):
    right = []

    for two in favorite:
        for one in basic:
            if two[0] == one[0] and two[1] > one[1]:
                time_part = (one[0], str(two[1] - one[1]).split(" ")[0],
two[1])
                right.append(time_part)


    print("[+] Finished!")
    return right

basic, favorite = decider(cleared_data)

content = get_object(one, two)
with open("C:\\....\\Result.csv", "w") as my_file:
    for line in content:
        my_file.writelines("{0},{1},{2}\n".format(line[0], line[1], line[2]))
```

V následující funkci s názvem „get_object“ porovnávám, jestli je „DateFrom“ z Oblíbeného tarifu větší než „DateFrom“ z prvního tarifu a spočítám počet dní mezi nimi. Pokud ano, uloží se mi do listu „right“. Data, která nesplňují tuto podmínku jsou zahozena a neukládám si je do žádného listu. Na závěr si uložím výsledné položky opět do .csv souboru. Výsledný soubor, který jsem upravila za pomocí Pythonu vypadá následovně.

```
Agenda,Days,Date
80,125,2018-02-05 00:00:00
96,378,2018-09-28 00:00:00
255,317,2018-10-11 00:00:00
270,398,2019-07-03 00:00:00
563,853,2020-01-04 00:00:00
651,1,2018-03-07 00:00:00
902,24,2017-10-06 00:00:00
1079,54,2017-09-24 00:00:00
1113,330,2018-06-27 00:00:00
1248,78,2018-05-15 00:00:00
1352,81,2019-06-20 00:00:00
1418,80,2018-11-24 00:00:00
1561,63,2017-10-03 00:00:00
```

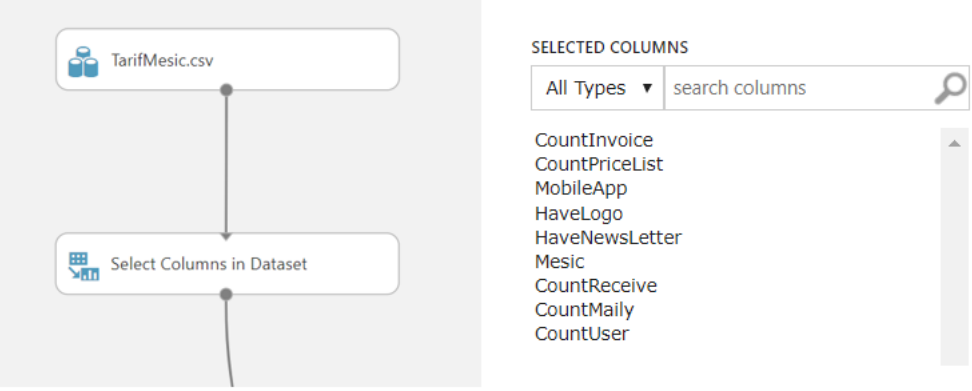
Obrázek 17: Výsledek po SQL dotazu

(Zdroj: Vlastní zpracování)

Posledním krokem v přípravě dat bylo převedení počtu dnů na měsíce, abych usnadnila modelu rozhodování, do které skupiny má danou Agendu zařadit. Dostačující bude určit měsíc, kdy by mohl přejít na vyšší tarif. Počet dnů jsem vydělila číslem 30 (průměrná délka měsíce) a zaokrouhlila nahoru, abych dostala pouze celá čísla. Soubor jsem nahrála do databáze, abych s ním mohla lépe manipulovat. Po nahrání souboru do databáze jsem ho propojila s ostatními tabulkami, ve kterých se nachází parametry, které budu používat při trénování modelu. Výsledný soubor má 3 143 řádků.

3.2.1.4 Výběr modelu

Další důležitý krok je správné vybrání typu učení, které budu používat. Výhodou je, že v Microsoft Azure Machine Learning Studiu se dá zároveň pracovat na stejných datech s více typy učení. Dokonce je možnost kombinovat samotné modely, tedy – regresi, klasifikaci a shlukování. Jelikož pracuji s predikcemi diskrétních hodnot, pro můj případ poslouží nejlépe klasifikační model. Potřebuji říct modelu, které parametry má použít a najít mezi nimi vztah, díky kterému je pak rozdělí do různých skupin, v mém případě jsou to skupiny měsíců neboli počet měsíců, za jak dlouhou dobu zákazník může přejít na vyšší tarif.



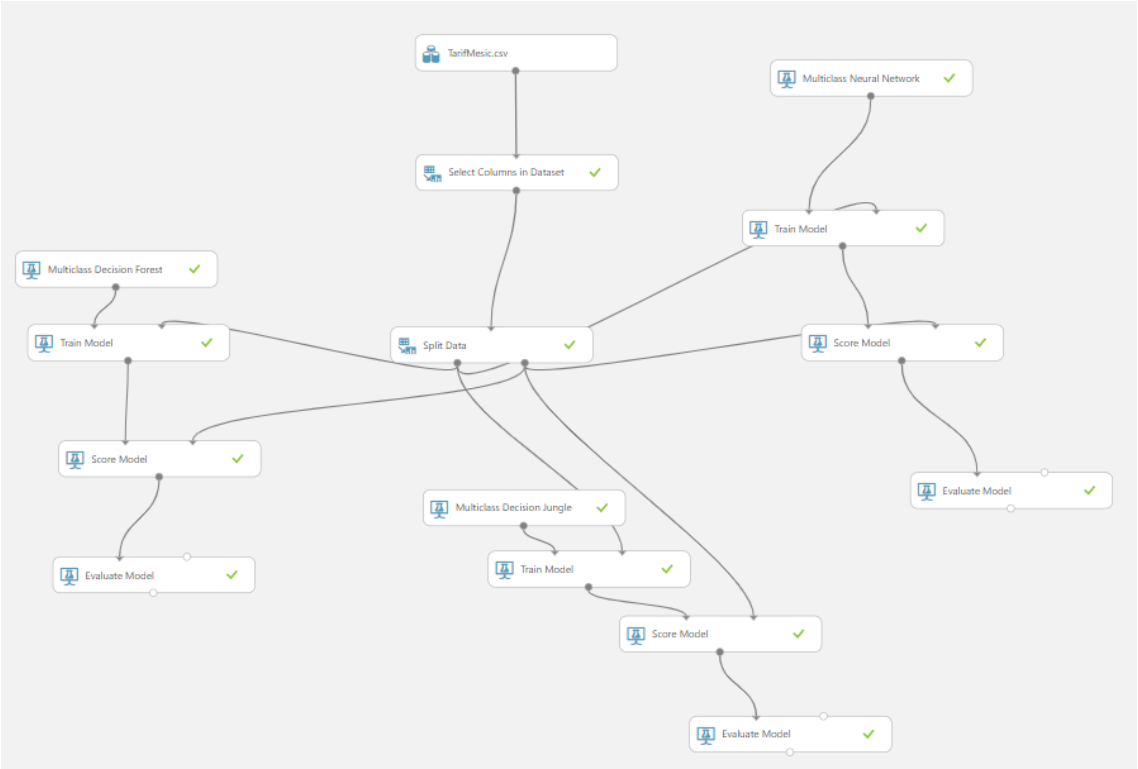
Obrázek 18: Azure ML Studio – nastavení datasetu

(Zdroj: Vlastní zpracování)

Do Machine Learning studia si nahraji mou datovou sadu, tedy soubor, z kterého chci čerpat data. Následně si vyberu sloupečky, které mě budou v predikci zajímat. Zahrnuji tam i sloupec Měsíc, který budu predikovat. V Machine Learning studiu je možné využít Spuštění Python skriptu přímo v něm. Takže Python skript, který jsem aplikovala v předchozím kroku bych mohla vložit přímo do tohoto studia. Ale z toho důvodu, že jsem dále se souborem musela pracovat a propojovat ho s ostatními tabulkami v databázi, tak pro mě bylo lepší vytvořit si skript v Pythonu zvlášť.

3.2.1.5 Trénování modelu

V Machine Learning Studiu použiji komponentu „Split Data“ a rozdělím si data na trénovací a testovací. 80 % dat jsem si nastavila jako trénovací, zbytek dat poslouží jako testovací sada. Po rozdělení dat už stačí jen přetáhnout do Studia modely, které chci použít a začít data trénovat. Výsledný model může vypadat následovně. V komponentě „Train Model“ si nastavím hodnotu, kterou chci predikovat. V mém případě to bude Měsíc.



Obrázek 19: Azure ML Studio – Trénování modelu

(Zdroj: Vlastní zpracování)

3.2.1.6 Vyhodnocení modelu

Data se natrénovala a teď je potřeba ověřit, jak dopadlo predikování na testovacích datech. Ve „Score modelu“ vidím pravděpodobnost vložení agendy do dané skupiny a k tomu výslednou hodnotu, do které byla Agenda vložena. Po otevření „Evaluate model“ vidím výsledky přesného odhadu modelu a matici záměny, která nám ve sloupcích ukazuje skutečnou hodnotu a v řádcích predikci. Přesnost modelu by se měla pohybovat nad 95 %. Zbytek je označováno jako statistická odchylka. Samozřejmě čím bude model přesnější tím lépe. V mém případě se průměrná přesnost pohybovala kolem 95 %.

	1	2	3	4	5	6
1	19.2%	15.2%	5.1%	12.1%	3.0%	5.1%
2	30.6%	16.1%	4.8%	16.1%	1.6%	3.2%
3	22.7%	11.4%	6.8%	4.5%		2.3%
4	23.0%	11.5%	9.8%	16.4%	6.6%	3.3%
5	25.0%	3.6%		10.7%	7.1%	10.7%

Obrázek 20: Azure ML Studio – vyhodnocení modelu

(Zdroj: Vlastní zpracování)

Jak už jsem psala výše sloupce jsou reálné hodnoty a řádky jsou predikované hodnoty. Diagonála na matici znázorňuje, kam by se predikce měla trefit. V tomto případě, kdy odhaduji časovou proměnnou je pro mě důležité, aby se model trefil co nejpřesněji kolem hlavní diagonály. Ideálně jedno políčko z každé strany diagonály. Model se přesněji trefoval v začátečních hodnotách, čím byly hodnoty vyšší, tím byly predikce méně přesné. Je to především z toho důvodu, že vysokých hodnot nebyl tak velký počet jako nízkých. Model tedy měl na nižších hodnotách větší šanci natrénovat si více možností.

Jednotlivé typy učení si můžeme nakonfigurovat podle potřeb a tím ovlivnit budoucí výsledky. U Multiclass Decision Forest a Multiclass Decision Jungle se rozhodujeme, zda použijeme metodu „Bagging“ nebo „Replicate“. V metodě Bagging každý strom roste na novém vzorku, který je vytvořený náhodně z původního datasetu. Výstupy modelů jsou průběžně agregované (20).

V metodě Replicate je každý strom trénován stále na těch stejných vstupních datech. Rozhodnutí, která data se použijí pro strom jsou náhodně generovaná (20).

Dále si definujeme mód trénování. Můžeme využít Single Parameter nebo Parameter Range. Rozdíl je v tom, že při Single Parameter zadáváme určitou hodnotu a při

Parameter Range zadáváme rozmezí, protože si nejsme jisti, která hodnota je ta nejvhodnější (20).

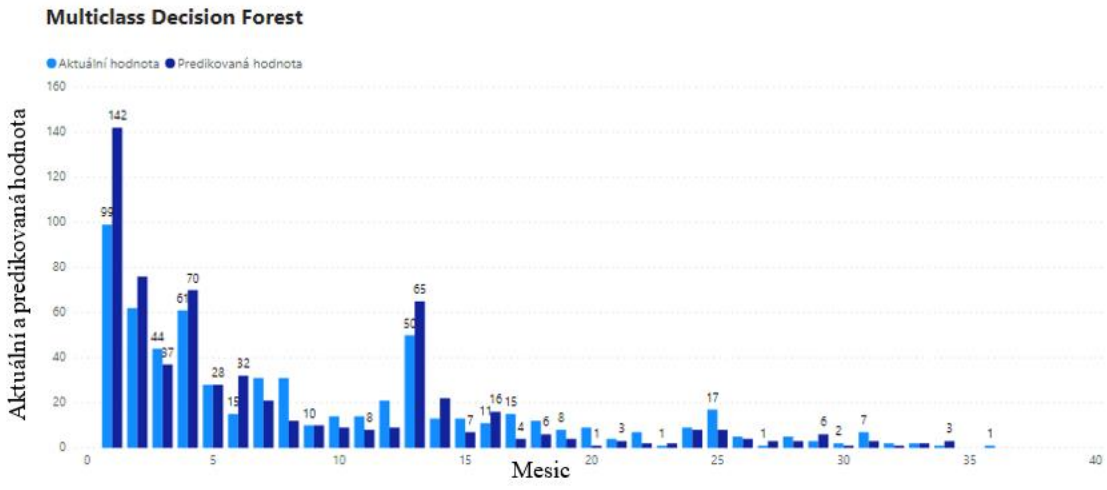
Hodnoty, které si nastavujeme jsou Number of decision trees, což nám udává maximální počet stromů, které model může vytvořit. Čím více použijeme stromů, tím můžeme získat lepší pokrytí možností dat, ale doba trénování se může prodloužit (20).

U algoritmu Multiclass Decision Jungle se používá pojem Number of decision DAGs a indikuje to počet grafů, které mohou být vytvořeny (21).

Multiclass Decision Forest

Model Multiclass Decision Forest má průměrnou přesnost 95,36 %. Na grafu níže jde vidět, že model neodhadl množství hodnot v prvním měsíci. Kdyby v prvním měsíci bylo méně predikovaných hodnot, model by se dal považovat za relativně přesný, jelikož další měsíce už mají podobnou tendenci jako aktuální hodnoty. Jde vidět, že první měsíce jsou v tomto modelu predikované více než reálně jsou. Nastavení modelu bylo následující:

- Resampling method: Bagging
- Create trainer mode: Single Parameter
- Number of decision trees: 64
- Maximum depth of the trees: 86



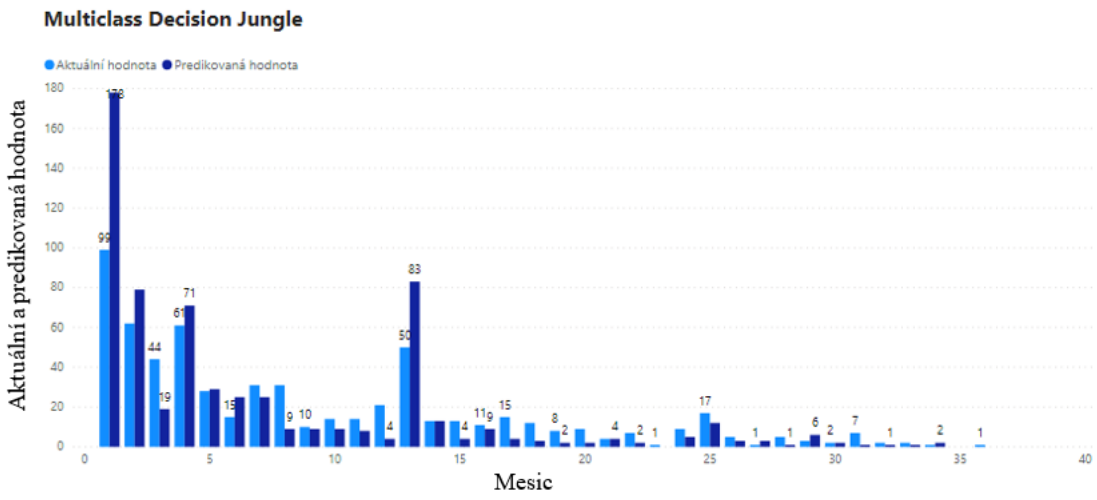
Graf 4: Multiclass Decision Forest

(Zdroj: Vlastní zpracování)

Multiclass Decision Jungle

Model Multiclass Decision Jungle má přesnost 95,39 %. Opět největší výskyt hodnot je u prvního měsíce, další výrazný nárůst je u měsíce 13., jinak jsou hodnoty podobné jako u modelu Multiclass Decision Forest.

- Resampling method: Bagging
- Create trainer mode: Single Parameter
- Number of decision DAGs: 64
- Maximum depth of the decision DAGs: 86

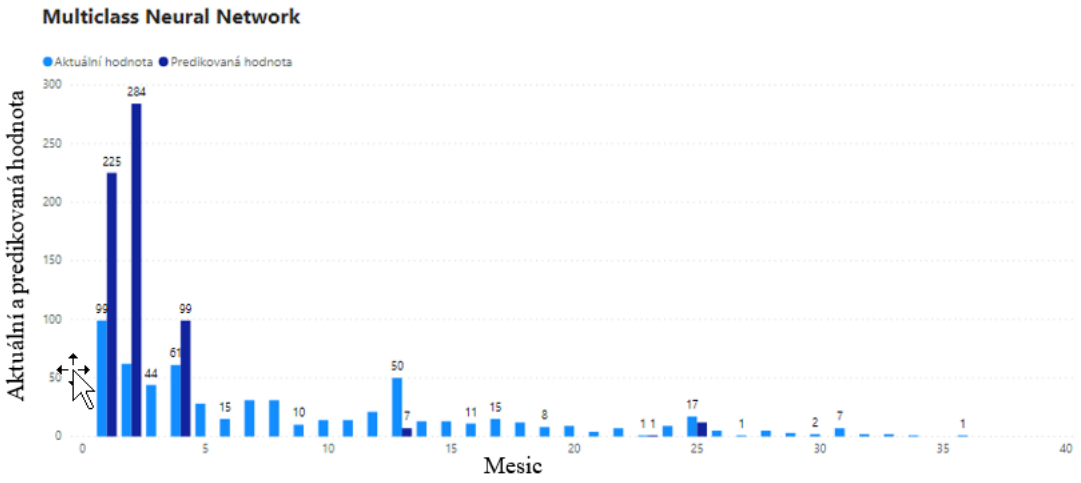


Graf 5: Multiclass Decision Jungle

(Zdroj: Vlastní zpracování)

Multiclass Neural Network

Model Multiclass Neural Network má přesnost 95,33 %, ale při pohledu na graf je patrné, že predikoval převážně na počáteční měsíce a dále už skoro vůbec. Tento model vychází ze všech nejhůře.



Graf 6: Multiclass Neural Network

(Zdroj: Vlastní zpracování)

Z grafů a vyhodnocení pomocí komponenty „Evaluate Model“ je patrné, že model Multiclass Neural Network nebudu vůbec používat, jelikož rozložení hodnot neodpovídá realitě. Horní dva modely vypadají, že by mohly predikovat přesnější hodnoty. V následujícím kroku se pokusím poměnit parametry a nastavení u daných modelů a podívám se, zda se mi průměrná přesnost a predikované hodnoty ještě zlepšily.

3.2.1.7 Ladění parametrů

U obou modelů jsem zvýšila počet stromů a maximum stromů. Zkusila jsem přidat nebo odebrat jiné parametry, ale nedosáhla jsem lepších výsledků. Parametry, které ovlivňují model nechávám stejné. Analyzovala jsem si znovu data, které mám v datasetu. Zjistila jsem, že četnost vyšších hodnot je nízká. Měsíc 45. je zastoupen v datasetu pouze jednou. Rozhodla jsem se, že hodnoty, které mají četnost pouze od 1 do 20 ze souboru vymažu. Týkalo se to měsíců 31 až 45.

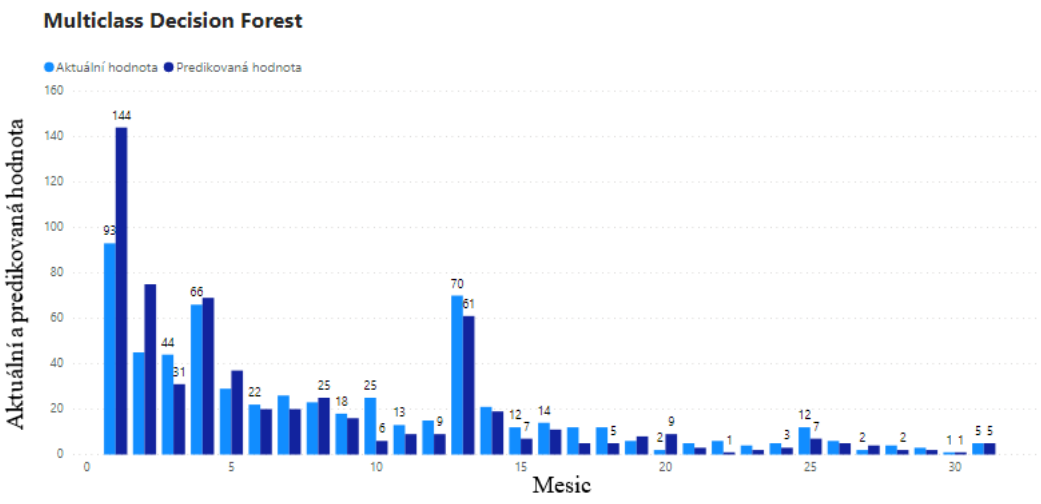
Nejlepšího výsledku, kterého se mi povedlo dosáhnout byl u Multiclass Decision Forest a to 95,40 %.

	1	2	3	4	5	6
1	33.3%	12.9%	5.4%	10.8%	5.4%	4.3%
2	24.4%	17.8%	8.9%	8.9%	6.7%	2.2%
3	31.8%	11.4%	9.1%	9.1%	4.5%	
4	15.2%	13.6%	6.1%	12.1%	6.1%	6.1%
5	20.7%	6.9%	3.4%	13.8%	6.9%	3.4%

Obrázek 21: Azure ML Studio – ladění parametrů

(Zdroj: Vlastní zpracování)

Jak je vidět na obrázku výše, hodnoty prvního měsíce byly trefovány s větší přesností, než tomu bylo u předchozího modelu. Na spodním grafu je vidět, že model lépe opisuje trend, i když jako v minulém případě první měsíc má vyšší četnost predikovaných hodnot než aktuálních hodnot.



Graf 7: Ladění parametrů – Multiclass Decision Forest

(Zdroj: Vlastní zpracování)

3.2.1.8 Predikce

Výsledná data je možné uložit si do souboru .csv a dále s nimi podle potřeby pracovat. Model lze také uložit jako webovou službu, přes kterou je možné nahrát svá aktuální data, která chceme predikovat.

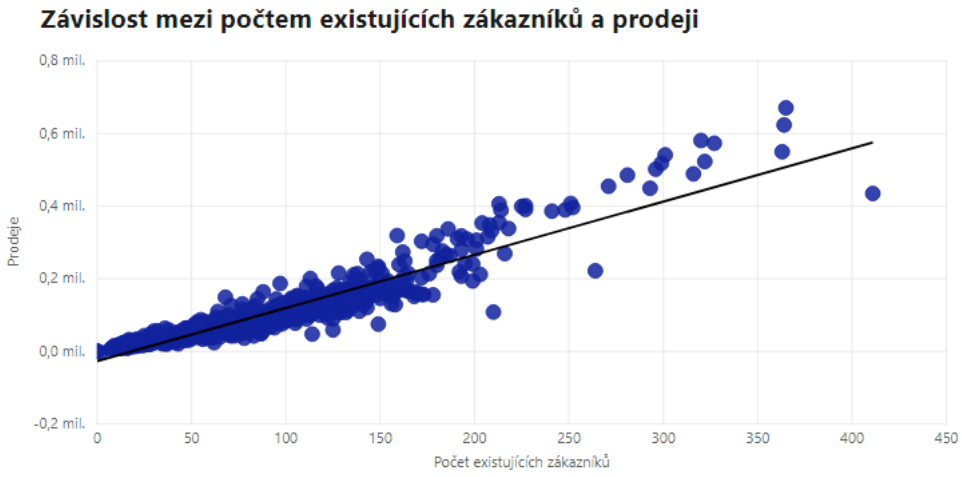
3.2.2 Druhý prediktivní model

3.2.2.1 Definice problému

Tato ukázka strojového učení se bude zabývat predikováním budoucích prodejů iDokladu.

3.2.2.2 Výběr dat

Pro tuto predikci jsem si vybrala prodeje iDokladu od roku 2017. Na datech je krásně vidět lineární závislost a nárůst prodejů v průběhu měsíců. Na tento typ predikce je vhodné použít regresní typ učení. Na začátku se musím zamyslet, která data mohou být na prodejích závislá. Jak jsem ukazovala na předchozí predikci, uchováváme údaje, kdy, který zákazník si zakoupil tarif a na jak dlouho. Díky tomu dokážeme už dopředu určit, kterého zákazníka v budoucnu budeme mít. Můžu si tedy rozdělit zákazníky na dvě sekce – na existující a nové. O nových samozřejmě dopředu nevíme, ale na základě předchozích zkušeností je to možné snadno odhadnout.



Graf 8: Lineární závislost mezi počtem existujících zákazníků a prodeji

(Zdroj: Vlastní zpracování)

Graf 9: Lineární závislost mezi počtem nových zákazníků a prodejů

(Zdroj: Vlastní zpracování)

Z grafů je patrné, že závislost mezi počtem existujících a nových zákazníků je velká. Dává smysl, že závislost mezi existujícími zákazníky je větší než mezi novými. Každý zákazník, kterého firma má vstupuje do proměnné prodejů.

3.2.2.3 Příprava dat

V tomto případě je příprava dat jednodušší. Spočítala jsem si počet existujících a nových zákazníků. Noví zákazníci jsou ti, kteří mají pouze jeden atribut DateFrom. Pokud už mají hodnot v tomto atributu více, tak jsou existující, protože už si službu iDoklad předplatili vícekrát.

3.2.2.4 Výběr modelu

Jak jsem psala výše, na tento problém je nejlepší využít regresní typ učení. Z předchozích grafů je patrné, že závislost mezi počty zákazníků a prodeji je lineární.

Budu se řídit podle následujícího vzorce na lineární regresi.

$$y = ax + b$$

3.2.2.5 Trénink modelu

V programovacím jazyku Python si vytvořím lineární model. Nejdříve si nahraji do Pythonu řadu knihoven, které se používají pro strojové učení. Opět si svá data nahraji

ze .csv souboru a do proměnné x, která je závislá, si uložím počty existujících a nových zákazníků. Do proměnné y si uložím prodeje iDokladu.

```
import pandas as pd
import numpy as np
import sklearn
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
import matplotlib.pyplot as plt
import datetime

dataset = pd.read_csv('SalesiDoklad.csv')
x = dataset.iloc[:, 1:3]
y = dataset['Sales']

x_train, x_test, y_train, y_test =
sklearn.model_selection.train_test_split(x, y, test_size=0.1)

lin = LinearRegression()
lin.fit(x_train, y_train)
```

Data si opět rozdělím na trénovací a testovací. Použiju funkci z importované knihovny – `sklearn.model_selection.train_test_split`. Nastavím si, že chci používat lineární regresi a následně začnu trénovat trénovací data pomocí funkce fit.

3.2.2.6 Vyhodnocení modelu

Kód zobrazený níže mi umožňuje na testovacích datech zjistit, jaká je úspěšnost modelu. Spočítám to pomocí funkce score, kterou si uložím do proměnné score.

```
score = lin.score(x_test, y_test)
print(score)
```

Pythonem si nechám zobrazit výsledné skóre. Jelikož trénovací data jsou vybírána náhodně, úspěšnost je pokaždé jiná.

```
print('Coefficient: \n', lin.coef_)
print('Intercept: \n', lin.intercept_)
```

Dále si nechám zobrazit koeficienty, které dále použiji při predikování. Výsledné hodnoty jsou následující.

Dosažené skóre (přesnost modelu): 0.9351755645709828

Coefficient: [-906.78312396 1553.16382387]

Intercept: -18024.608464666206

3.2.2.7 Ladění parametrů

Abych získala, co největší přesnost mého modelu musím najít nejvhodnější trénovací data, s největší přesností, které mi potom předají nejvhodnější koeficienty, s kterými mohu dále pracovat.

```
high = 0
for _ in range(100):
    x_train, x_test, y_train, y_test =
sklearn.model_selection.train_test_split(x, y, test_size=0.1)

    lin = LinearRegression()
    lin.fit(x_train, y_train)

    score = lin.score(x_test, y_test)

    if score > high:
        high = score

        with open("coef.pickle", "wb") as f:
            pickle.dump(lin, f)

pickle_in = open("coef.pickle", "rb")
lin = pickle.load(pickle_in)
print(high)
print('Coefficient: \n', lin.coef_)
print('Intercept: \n', lin.intercept_)
```

Na to použiji for cyklus v Pythonu, který mi bude hledat v trénovacích datech nejpřesnější hodnoty. Na začátek si nadefinuji novou proměnnou high s hodnotou nula. Ve for cyklu si data opět rozdělím na testovací a trénovací a použiji lineární model. Pokud proměnná score, která značí přesnost mého modelu, bude větší než proměnná high, tak se uloží do proměnné high. Zároveň si nejlepší model uložím do pickle souboru. Po nalezení nejvhodnějšího modelu si nechám opět zobrazit data.

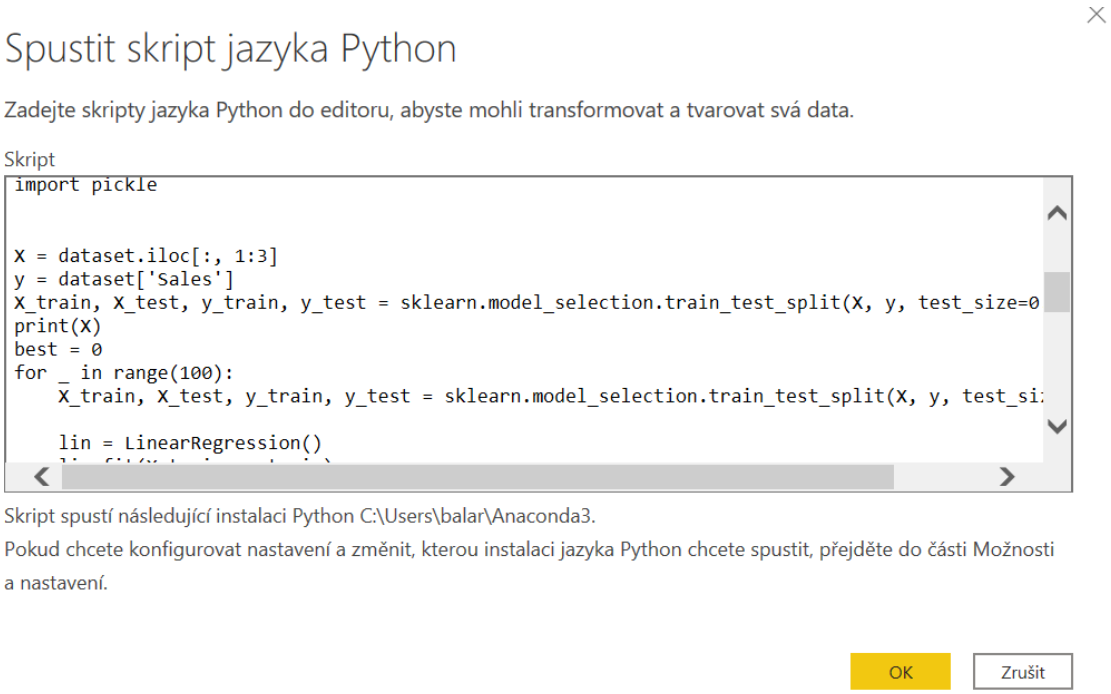
Dosažené skóre (přesnost modelu): 0.9496051325348878

Coefficient: [-805.56871197 1546.20154007]

Intercept: -19201.56574176204

3.2.2.8 Predikce

Model jsem se rozhodla využívat v rámci reportu v programu MS Power BI, které se běžně používají ve společnosti.



Obrázek 22: Spuštění Pythonu v Power BI

(Zdroj: Vlastní zpracování)

Přímo z Power BI je možné spustit Python skript. Překopíruji tedy celý můj skript do určeného okna. Z mého skriptu odmažu proměnnou dataset. Proměnná dataset je označená tabulka v Power BI, nad kterou právě spouštím skript. Po spuštění skriptu se tato tabulka smaže a nahradí se to mou tabulkou s názvem parametru (CountNew, CountExisting a Intercept) a hodnotou koeficientů. Abych mohla takto nahrát do Power BI mé hodnoty, sbalím si je v Pythonu do balíčku následujícím příkazem:

```
coefficient = pd.DataFrame(zip(x.columns, lin.coef_), columns=('name',
'coefficient'))
coefficient = coefficient.append({'name': 'intercept', 'coefficient':
lin.intercept_}, ignore_index=True)
```

Výsledkem je tabulka s názvy proměnných a jejími koeficienty.

name	coefficient
CountNew	-806,57
CountExisting	1546,2
Intercept	-19201,57

Obrázek 23: Výsledné hodnoty pomocí lineární regrese

(Zdroj: Vlastní zpracování)

Zjistím počet existujících zákazníků za označený měsíc. Odchytím si minimální a maximální hodnotu označené části na časové ose. Dostanu první a poslední den v měsíci, přičemž mě zajímá počet existujících zákazníků mezi těmito dvěma hodnotami. Hodnoty nových zákazníků nám nejsou předem známé. Ale manažer, který bude s reportem pracovat, může na základě svých zkušeností odhadnout nárůst zákazníků nebo je možné řídit se podle nárůstu v minulých měsících. Jelikož počet nových zákazníků se bude zadávat jako vlastní hodnota, vytvořím si v Power BI vlastní parametr. Parametr se uloží jako tabulka s hodnotou, která se bude měnit na základě hodnoty, kterou zadá uživatel do textového pole.

Výpočet predikce jsem provedla v jazyku DAX a použila jsem koeficienty, které jsem získala z Python skriptu.

```
Predict = var new =
CALCULATE(sum(CoefSales[coefficient]);CoefSales[name]="CountNew")
var exist =
CALCULATE(SUM(CoefSales[coefficient]);CoefSales[name]="CountExisting")
var inter =
CALCULATE(SUM(CoefSales[coefficient]);CoefSales[name]="Intercept")
var result = inter+((exist*[CountExist])+(new*NewCustomer[Hodnota
NewCustomer]))
return result
```

Interaktivní report s ručním zadáváním počtu nových zákazníků a vypočteným počtem existujících zákazníků vypadá následovně:

Obrázek 24: Predikce prodejů v Power BI – ruční zadávání zákazníků

(Zdroj: Vlastní zpracování)

Manažer, který bude report využívat si může zapnout možnost spočítat nové zákazníky dle minulého roku. Nárůst zákazníků se může v rámci různých obdobích opakovat. Můžeme předpokládat, že v letních měsících, kdy lidé spíše odpočívají a jsou na dovolené, zákazníků přibývá méně. Tato možnost je hlavně pro uživatele, kteří nemají takový přehled o tom, jak se společnosti aktuálně daří a jaká je poptávka na trhu. Bude pro ně dostačující počítat predikce dle minulého roku.

Na výpočet počtu nových zákazníků v minulých letech použiju funkci Power BI s názvem SAMEPERIODLASTYEAR(). Funkce se podívá zpět o rok přesně v ohraničeném období (v mém případě v měsících) a v tom období spočítá aktuální počet nových zákazníků.

Obrázek 25: Predikce prodejů v Power BI – zákazníci dle minulého roku

(Zdroj: Vlastní zpracování)

3.3 Zhodnocení vlastního návrhu řešení

3.3.1 Ekonomické zhodnocení

Do ekonomického zhodnocení jsem zahrnula platy zaměstnanců, kteří se budou na projektu podílet. Tyto představují největší náklad na tento projekt. Dále musím započítat licenci na Microsoft Azure Machine Learning Studio, kterou budou developeři využívat pro své predikce.

BI developer na první predikci tarifů iDokladu – čistý časový odhad: 106,25 hodin, časový odhad na dny: 17 dnů, částka: 38 173 Kč

BI developer na druhou predikci prodejů iDokladu – čistý časový odhad: 76,25 hodin, časový odhad na dny: 13 dnů, částka: 29 120 Kč

Projektový manažer – čistý časový odhad: 6 hodin, časový odhad na dny: 1 den, částka: 3 500 Kč

Obchodní manažer –časový odhad: 5,83 hodin, časový odhad na dny: 1 den, částka: 3 250 Kč

Marketingový manažer – 2 186 Kč, časový odhad: 5,83 hodin, časový odhad na dny: 1 den, částka: 3 000 Kč

Azure Machine Learning Studio - €8,425 (230,09 Kč) za pracovní prostor ML Studio/měsíc

€0,844 (23 Kč) za hodinu experimentů (Studio)

3.3.2 Přínosy řešení

Hlavní přínos práce spočívá v tom ukázat možnosti, jak využít strojové učení ve zvolené firmě. Rozhodla jsem se proto udělat dva rozdílné modely i s různým využitím technologií. První model je, co se týče přípravy dat a zamyšlení se nad spojitostmi mezi daty náročnější. Samotný výběr, které parametry mohou být pro model relevantní, bylo obtížné vybrat. Na tomto případě jsem ukázala, jak je důležité připravit si a dobře vybrat svá data. Při samotné přípravě bude potřebné využít i jiné technologie než jen jazyk SQL, v mém případě jsem si pomohla programovacím jazykem Python. Samotný proces učení a nastavování v Azure Machine Learning Studio bylo o zkoušení vhodných typů učení a nastavování dobrých parametrů.

Druhý model jsem pojala opět jiným způsobem. Zaměřila jsem se na predikci, kterou mohu ihned zobrazit v reportovacím nástroji Power BI. Jelikož se ve firmě s tímto nástrojem hojně pracuje je vhodné využít propojení s programovacím jazykem Python, ve kterém si připravím prediktivní model a výsledky si zobrazím rovnou v Power BI. Cílem bylo ukázat, jak se mohou tyto dvě technologie vzájemně doplňovat.

Hlavní přínosy dosažené tímto projektem:

- efektivnější práce s firemními daty,
- kvalitnější predikce,
- automatizovaný systém predikování,
- úspora času,
- potenciál získání nových zákazníků pro placené produkty.

ZÁVĚR

Cílem mé diplomové práce bylo analyzovat současný stav organizace a její produkty a na základě analýzy vyhodnotit mezery ve využívání dat. Hlavním cílem bylo pomocí strojového učení přinést firmě nové informace o jejich zákaznících.

Při vytváření modelů jsem se snažila využít více možných typů strojového učení a představit technologie, díky kterým se dá dosáhnout predikcí. Představila jsem komplexní postup při vytváření modelu v Azure ML Studio i se samotnou přípravou dat v programovacím jazyku Python. V druhém modelu jsem kromě Pythonu použila také reportovací nástroj Power BI a ukázala, jak automatizovaně se dá tento model používat.

Závěrem mohu říct, že se mi cíl, který jsem si zvolila podařilo naplnit.

SEZNAM ZDROJŮ

(1) Databáze a jazyk SQL. *Interval.cz* [online]. © 2000 [cit. 2020-01-10]. Dostupné z: https://www.interval.cz/clanky/databaze-a-jazyk-sql/

(2) BEN-GAN, Itzik, Dejan SARKA and Ron TALMAGE. *Querying Microsoft SQL Server 2012: Exam 70-461 Training Kit*. 1st Edition. California: Microsoft, 2012. ISBN 978-0-7356-6605-4.

(3) Hlavní principy datových skladů a proces jejich vytváření. *Systemonline.cz* [online]. ©2000 [cit. 2020-01-10]. Dostupné z: https://www.systemonline.cz/clanky/hlavni-principy-datovych-skladu-a-proces-jejich-vytvareni.htm

(4) LABERGE, Robert. *Datové sklady: agilní metody a business intelligence*. 1.vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3729-1.

(5) Fakta a dimenze – Tabulky v datovém skladu. *Biportal.cz* [online]. © 2018 [cit. 2020-01-10]. Dostupné z: https://biportal.cz/fakta-dimenze-tabulky-v-datovem-skladu/

(6) Star and SnowFlake Schema in Data Warehouse. *Guru99.com* [online]. [cit. 2020-01-15]. Dostupné z: https://www.guru99.com/star-snowflake-data-warehousing.html

(7) SSIS | Integration Services pro začátečníky – Úvod, BIDS, Project, Package, SSIS Toolbox. *Biportal.cz* [online]. © 2019 [cit. 2020-01-15]. Dostupné z: https://biportal.cz/ssis-integration-services-pro-zacatecniky-predstaveni/

(8) Business Intelligence. *Managementmania.com* [online]. [cit. 2020-01-15]. Dostupné z: https://managementmania.com/cs/business-intelligence

(9) Machine Learning for Beginners. *Towardsdatascience.com* [online]. © 2018 [cit. 2020-04-12]. Dostupné z: https://towardsdatascience.com/machine-learning-for-beginners-d247a9420dab

(10) SLEPT. *Ict-123.com* [online]. [cit. 2020-04-28]. Dostupné z: http://www.ict-123.com/Metody/SLEPT

(11) McKinsey 7S. *Managementmania.com* [online]. [cit. 2020-04-28]. Dostupné z: https://managementmania.com/cs/mckinsey-7s

(12) Analýza pěti sil 5F (Porter's Five Forces). *Managementmania.com* [online]. [cit. 2020-04-28]. Dostupné z: https://managementmania.com/cs/analyza-5f

(13) SWOT analýza. *Managementmania.com* [online]. [cit. 2020-04-28]. Dostupné z: https://managementmania.com/cs/swot-analyza

(14) Společnost. *Solitea* [online]. ©2020 [cit. 2020-04-10]. Dostupné z: https://solitea.cz/spolecnost/

(15) Produkty. *Solitea* [online]. ©2020 [cit. 2020-04-10]. Dostupné z: https://solitea.cz/produkty/

(16) Ceník. *iDoklad* [online]. ©2020 [cit. 2020-04-10]. Dostupné z: https://www.idoklad.cz/cenik

(17) MÜLLER, Andreas C. a Sarah GUIDO. *Introduction to machine learning with Python: a guide for data scientists.* 1 st Edition. Sebastopol, CA: O'Reilly Media, 2016. ISBN 9781449369415.

(18) My first experience with Microsoft Azure Machine Learning Studio. *Towardsdatascience.com* [online]. ©2019 [cit. 2020-05-22]. Dostupné z: https://towardsdatascience.com/my-first-experience-with-microsoft-azure-machine-learning-1f054d252808

(19) The 7 Steps Of Machine Learning. *Towardsdatascience.com* [online]. ©2017 [cit. 2020-05-22]. Dostupné z: https://towardsdatascience.com/the-7-steps-of-machine-learning-2877d7e5548e

(20) Multiclass Decision Forest module. *Docs.microsoft.com* [online]. ©2020 [cit. 2020-05-22]. Dostupné z: https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/multiclass-decision-forest

(21) Multiclass Decision Jungle. *Docs.microsoft.com* [online]. © 2019 [cit. 2020-05-22]. Dostupné z: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/multiclass-decision-jungle

SEZNAM ZKRATEK

SQL – Structured Query Language

T-SQL – Transact Structured Query Language

API – Application Programming Interface

BI – Business Intelligence

DWH – Data Warehouse

ML – Machine Learning

SEZNAM GRAFŮ

SEZNAM OBRÁZKŮ

SEZNAM TABULEK